

#4

IN THE U.S. PATENT AND TRADEMARK OFFICE

In Re Application of:

Shuichi Takayama et al.

Serial No.: 09/301,284

Filed: April 28, 1999

For: PROCESSOR FOR EXECUTING
INSTRUCTIONS IN UNITS THAT ARE
UNRELATED TO THE UNITS IN WHICH
INSTRUCTIONS ARE READ AND A
COMPILER, AN OPTIMIZATION
APPARATUS, AN ASSEMBLER, A LINKER, A
DEBUGGER, AND A DISASSEMBLER FOR
SUCH PROCESSOR



Examiner:

Group Art Unit:

Irvine, CA 92614

June 1, 1999

LETTER

Assistant Commissioner for Patents
Washington, DC 20231

Dear Sir:

In response to the Notice to File Missing Parts, attached is the Declaration and Power of Attorney for filing in the above-identified case together with a check for \$130.00 to cover the fee. Also attached is the priority document, Japan 10-118326, in accordance with 35 USC § 119.

If there are any questions, please do not hesitate to contact the undersigned attorney.

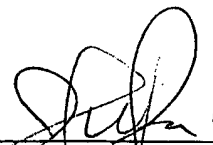
I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to Assistant Commissioner for Patents, Washington, DC 20231, on June 1, 1999, by Karen L. Miyakawa

Very truly yours,

PRICE, GESS & UBELL


Signature

June 1, 1999
Date of Signature


Joseph W. Price, Reg. No. 25,124
2100 S.E. Main St., Ste. 250
Irvine, CA 92614
949/261-8433



J.W. Price
949/261.8433
Shuichi Takayama et al,
S.N. 09/301,284

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

NAK-1-3686

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

1998年 4月28日

出願番号
Application Number:

平成10年特許願第118326号

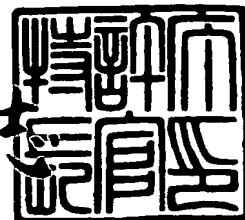
出願人
Applicant(s):

松下電器産業株式会社

1999年 3月12日

特許庁長官
Commissioner,
Patent Office

伴佐山建志



出証番号 出証特平11-3013928

【書類名】 特許願

【整理番号】 5037790361

【提出日】 平成10年 4月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/45

【発明の名称】 コンパイラ、命令列最適化装置、アセンブラ、リンカ、デバッガ、逆アセンブラ及びプロセッサ

【請求項の数】 42

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 高山 秀一

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 小川 一

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 川口 謙一

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 檜垣 信生

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式会社内

【氏名】 小谷 謙介

【発明者】

【住所又は居所】 大阪府門真市大字門真 1006 番地 松下電器産業株式会社内

【氏名】 田中 哲也

【発明者】

【住所又は居所】 大阪府門真市大字門真 1006 番地 松下電器産業株式会社内

【氏名】 宮地 信哉

【発明者】

【住所又は居所】 大阪府門真市大字門真 1006 番地 松下電器産業株式会社内

【氏名】 瓶子 岳人

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100078204

【弁理士】

【氏名又は名称】 滝本 智之

【選任した代理人】

【識別番号】 100097445

【弁理士】

【氏名又は名称】 岩橋 文雄

【手数料の表示】

【予納台帳番号】 011305

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

特平 10-118326

【物件名】 要約書 1

【包括委任状番号】 9702380

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンパイラ、命令列最適化装置、アセンブラ、リンカ、デバッガ、逆アセンブラ及びプロセッサ

【特許請求の範囲】

【請求項 1】 命令あるいは分割された命令の一部を 2 の中乗数ではない複数個含む命令パケットのうち、実行すべき命令またはその一部を含む命令パケットを識別するパケットアドレスを保持する第一プログラムカウンタ手段と、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中で、実行すべき命令を特定するパケット内アドレスを保持する第二プログラムカウンタ手段と、前記第二プログラムカウンタ手段が保持するパケット内アドレスを、次に実行すべき命令を特定するパケット内アドレスに更新し、さらに次に実行すべき命令またはその一部が、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットに含まれない場合は、前記第一プログラムカウンタ手段が保持するパケットアドレスを次に実行すべき命令を含む命令パケットのパケットアドレスに更新するプログラムカウンタ更新手段と、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中の命令のうち、前記第二プログラムカウンタ手段が保持するパケット内アドレスによって特定される命令を実行する実行手段とを備えることを特徴とするプロセッサ。

【請求項 2】 前記命令パケットは固定の長さを有し、前記プログラムカウンタ更新手段は、前記第二プログラムカウンタ手段が保持するパケット内アドレスを、次に実行すべき命令を特定するパケット内アドレスに更新し、さらに次に実行すべき命令が、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットに含まれない場合は、前記第一プログラムカウンタ手段が保持するパケットアドレスを命令パケットの長さで決まる大きさだけ増加することを特徴とする請求項 1 記載のプロセッサ。

【請求項 3】 前記プロセッサはさらに、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中の命令のうち、前記第二プログラムカウンタ手段が保持するパケット内アドレスによって特定さ

れる命令からプログラムカウンタ相対値を検出するデコード手段と、前記第一プログラムカウンタ手段が保持するパケットアドレス及び前記第二プログラムカウンタ手段が保持するパケット内アドレスで構成する値と前記デコード手段が検出したプログラムカウンタ相対値とを加算する加算手段とを備え、前記実行手段は、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中の命令のうち、前記第二プログラムカウンタ手段が保持するパケット内アドレスによって特定される命令を前記加算手段の結果を用いて実行することを特徴とする請求項 1 又は 2 記載のプロセッサ。

【請求項 4】 前記プロセッサはさらに、前記第一プログラムカウンタ手段が保持するパケットアドレスと前記第二プログラムカウンタ手段が保持するパケット内アドレスとによって特定される命令と並列実行可能な単一あるいは複数の命令を一つの実行単位として特定する発行手段と、前記発行手段が実行単位として特定した命令から実行単位の先頭にある命令を基準とするプログラムカウンタ相対値を検出するデコード手段と、前記第一プログラムカウンタ手段が保持するパケットアドレス及び前記第二プログラムカウンタ手段が保持するパケット内アドレスで構成する値と前記デコード手段が検出したプログラムカウンタ相対値とを加算する加算手段とを備え、前記実行手段は、前記加算手段の結果を用いて前記発行手段が実行単位として特定した単一あるいは複数の命令を実行することを特徴とする請求項 1 又は 2 記載のプロセッサ。

【請求項 5】 前記プロセッサはさらに、前記第一プログラムカウンタ手段が保持するパケットアドレスと前記第二プログラムカウンタ手段が保持するパケット内アドレスとを実行すべき命令の前記実行単位中の偏位だけ補正するプログラムカウンタ補正手段を備え、前記デコード手段は前記発行手段が実行単位として特定した命令からこの命令を基準とするプログラムカウンタ相対値を検出し、前記加算手段は前記プログラムカウンタ補正手段が補正したパケットアドレス及び前記プログラムカウンタ補正手段が補正したパケット内アドレスで構成する値と前記デコード手段が検出したプログラムカウンタ相対値とを加算することを特徴とする請求項 4 記載のプロセッサ。

【請求項 6】 前記加算手段は、前記第一プログラムカウンタ手段が保持する

パケットアドレスと前記デコード手段が検出したプログラムカウンタ相対値の上位ビットとを加算する第一加算手段と、前記第二プログラムカウンタ手段が保持するパケット内アドレスと前記デコード手段が検出したプログラムカウンタ相対値の下位ビットとを加算する第二加算手段とからなることを特徴とする請求項3又は請求項4記載のプロセッサ。

【請求項7】 前記加算手段は、前記第一プログラムカウンタ手段が保持するパケットアドレスを前記プログラムカウンタ補正手段が補正したパケットアドレスと前記デコード手段が検出したプログラムカウンタ相対値の上位ビットとを加算する第一加算手段と、前記第二プログラムカウンタ手段が保持するパケット内アドレスを前記プログラムカウンタ補正手段が補正したパケット内アドレスと前記デコード手段が検出したプログラムカウンタ相対値の下位ビットとを加算する第二加算手段とからなることを特徴とする請求項5記載のプロセッサ。

【請求項8】 前記第一加算手段は、さらに前記第二加算手段で生じた桁上り数をも加算することを特徴とする請求項6又は請求項7記載のプロセッサ。

【請求項9】 前記加算手段は、前記第一プログラムカウンタ手段が保持するパケットアドレスと前記デコード手段が検出したプログラムカウンタ相対値の上位ビットとを加算する第一加算手段と、前記デコード手段が検出したプログラムカウンタ相対値の下位ビットを結果とする第二加算手段とからなることを特徴とする請求項3又は請求項4又は請求項5記載のプロセッサ。

【請求項10】 前記加算手段は、前記第二プログラムカウンタ手段が保持するパケット内アドレスを下位ビットとし前記第一プログラムカウンタ手段が保持するパケットアドレスを上位ビットとする値と前記デコード手段が検出したプログラムカウンタ相対値とを上位ビットと下位ビットとの区別なしに加算することを特徴とする請求項3又は請求項4又は請求項5記載のプロセッサ。

【請求項11】 前記デコード手段は、加算命令または減算命令中に指定されたプログラムカウンタ相対値を検出することを特徴とする請求項6から請求項10のいずれか1項に記載のプロセッサ。

【請求項12】 複数のリロケータブルコードを結合する結合手段と、前記複数のリロケータブルコードの中から2つの命令間のアドレスの差分に解決すべき

再配置情報を検出する検出手段と、前記2つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算する下位減算手段と、前記2つの命令について、前記一方のアドレスの上位ビットから前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出した再配置情報を前記算出手段で算出した値で解決する解決手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とするリンク。

【請求項13】 前記上位減算手段は、さらに前記下位算出手段で生じた桁借り数をも減算することを特徴とする請求項12記載のリンク。

【請求項14】 前記下位減算手段は、前記2つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算することに代えて、前記2つの命令の一方のアドレスの下位ビットを結果とすることを特徴とする請求項12記載のリンク。

【請求項15】 命令列をリロケータブルコードに変換する変換手段と、前記命令列の中から2つの命令間のアドレスの差分に解決すべきラベルを検出する検出手段と、前記2つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算する下位減算手段と、前記2つの命令について、前記一方のアドレスの上位ビットから前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出したラベルを前記算出手段で算出した値で解決する解決手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とするアセンブラ。

【請求項16】 前記上位減算手段は、さらに前記下位算出手段で生じた桁借

り数をも減算することを特徴とする請求項 15 記載のアセンブラ。

【請求項 17】 前記下位減算手段は、前記 2 つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算することに代えて、前記 2 つの命令の一方のアドレスの下位ビットを結果とすることを特徴とする請求項 15 記載のアセンブラ。

【請求項 18】 命令列の中から 2 つの命令間のアドレスの差分に解決すべきラベル及び命令のアドレスに解決すべきラベルを検出する検出手段と、前記命令列の各命令にアドレスを付与するアドレス付与手段と前記 2 つの命令について、前記アドレス付与手段が付与した一方のアドレスの下位ビットから前記アドレス付与手段が付与した他方のアドレスの前記下位ビットを減算する下位減算手段と、前記 2 つの命令について、前記アドレス付与手段が付与した前記一方のアドレスの上位ビットから前記アドレス付与手段が付与した前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出したラベルを前記算出手段で算出した値に変換し、前記ラベルを有する命令を前記算出手段で算出した値に応じた命令に変換する変換手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とする命令列最適化装置。

【請求項 19】 前記上位減算手段は、さらに前記下位減算手段で生じた桁借り数をも減算することを特徴とする請求項 18 記載の命令列最適化装置。

【請求項 20】 前記下位減算手段は、前記 2 つの命令について、前記アドレス付与手段が付与した一方のアドレスの下位ビットから前記アドレス付与手段が付与した他方のアドレスの前記下位ビットを減算することに代えて、前記 2 つの命令の一方のアドレスの下位ビットを結果とすることを特徴とする請求項 18 記載の命令列最適化装置。

【請求項 21】 オブジェクトコードの中に存在する命令ビットパターンを識別するためのアドレスの下位ビットと、前記命令ビットパターン中に指定される

相対アドレス値の下位ビットとを加算する下位加算手段と、前記命令ビットパターンを識別するためのアドレスの上位ビットと、前記相対アドレス値の上位ビットとを加算する上位加算手段と、前記上位加算手段が加算した結果が上位ビットであり、前記下位加算手段が加算した結果が下位ビットである値を算出する算出手段と、前記値とラベル名とを対にして記憶する記憶手段と、前記算出手段が算出した結果に基づいて前記記憶手段を検索してラベル名を得る検索手段と、逆アセンブル時に、前記検索手段が検索したラベル名を表示する表示手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令バケットを識別し、アドレスの前記下位ビットによって、命令バケット中の命令を特定することを特徴とする逆アセンブラ装置。

【請求項22】 前記上位加算手段は、さらに前記下位加算手段で生じた桁上がり数をも加算することを特徴とする請求項21記載の逆アセンブラ装置。

【請求項23】 前記下位加算手段は、オブジェクトコードの中に存在する命令ビットパターンを識別するためのアドレスの下位ビットと、前記命令ビットパターン中に指定される相対アドレス値の下位ビットとを加算することに代えて、前記相対アドレス値の下位ビットを結果とすることを特徴とする請求項21記載の逆アセンブラ装置。

【請求項24】 命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令バケットを識別するバケットアドレスを上位、前記命令バケットの中で実行すべき命令を特定するバケット内アドレスを下位とするプログラムカウンタ値と、プログラムカウンタ値の相対値とを、前記上位と前記下位に分離して加算する命令を生成するプログラムカウンタ値加算命令生成手段を備えることを特徴とするコンパイラ。

【請求項25】 命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令バケットを識別するバケットアドレスを上位、前記命令バケットの中で実行すべき命令を特定するバケット内アドレスを下位とするプログラムカウンタ値と、プログラムカウンタ値の相対値とを、前記下位の加算を行ない、その桁上がりを含めて前記上位の加算を行なう命令を生成するプログラムカウンタ値加算命令生成手段を備えることを特徴とするコンパイラ。

【請求項 26】 命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別するパケットアドレスを上位、前記命令パケットの中で実行すべき命令を特定するパケット内アドレスを下位とするプログラムカウンタ値と、プログラムカウンタ値の相対値とを、前記上位の加算を行ない、前記下位は前記相対値の前記下位の値とする命令を生成するプログラムカウンタ値加算命令生成手段を備えることを特徴とするコンパイラ。

【請求項 27】 命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットに含まれる対象命令のパケットアドレスを特定する命令特定手段と、前記命令特定手段が特定したパケットアドレスによって識別される命令を含む命令パケットのパケットアドレスを特定する命令パケット特定手段と、前記命令パケット特定手段が特定するパケットアドレスによって識別される命令パケットのビットパターンを検出するビットパターン検出手段と、前記ビットパターン検出手段が検出したビットパターンにおける前記対象命令に相当するビット列を、該ビット列と異なるビット列に更新するビットパターン更新手段とを備えることを特徴とするデバッガ。

【請求項 28】 上位ビットによって、命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2 つの前記命令間の前記アドレスの差分に解決すべき再配置情報を、2 つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で、解決する再配置情報解決ステップとをリンカに実行させるためのプログラムを記録した記録媒体。

【請求項 29】 前記再配置情報解決ステップは、さらに前記下位ビットの減算によって生じる桁借り数をも前記上位ビットから減算することを特徴とする請求項 28 記載の記録媒体。

【請求項 30】 前記再配置情報解決ステップは、2 つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算す

ることに代えて、2つの前記命令の一方の前記アドレスの前記下位ビットを結果とすることを特徴とする請求項 28 記載の記録媒体。

【請求項 31】 上位ビットによって、命令あるいは分割された命令の一部を 2 の中乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2つの前記命令間の前記アドレスの差分に解決すべきラベルを、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で、解決するラベル解決ステップとをアセンブラに実行させるためのプログラムを記録した記録媒体。

【請求項 32】 前記ラベル解決ステップは、さらに前記下位ビットの減算によって生じる桁借り数をも前記上位ビットから減算することを特徴とする請求項 31 記載の記録媒体。

【請求項 33】 前記ラベル解決ステップは、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算することに代えて、2つの前記命令の一方の前記アドレスの前記下位ビットを結果とすることを特徴とする請求項 31 記載の記録媒体。

【請求項 34】 上位ビットによって、命令あるいは分割された命令の一部を 2 の中乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2つの前記命令間の前記アドレスの差分に解決すべきラベルを、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で変換し、前記ラベルを有する命令を前記値に応じた命令に変換する第一命令変換ステップと、前記アドレスに解決すべきラベルを有する命令に対して、前記ラベルを前記命令アドレス決定ステップが付加したアドレスに変換し、このアドレスに応じた命令に変換する第 2 命令変換ステップとを命令列最適化装

置に実行させるためのプログラムを記録した記録媒体。

【請求項 35】 前記第一命令変換ステップは、さらに前記下位ビットの減算によって生じる桁借り数をも前記上位ビットから減算することを特徴とする請求項 34 記載の記録媒体。

【請求項 36】 前記第一命令変換ステップは、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算することに代えて、2つの前記命令の一方の前記アドレスの前記下位ビットを結果とすることを特徴とする請求項 34 記載の記録媒体。

【請求項 37】 命令あるいは分割された命令の一部を2の中乗数ではない複数個含む命令パケットのうち、実行すべき命令またはその一部を含む命令パケットを識別するパケットアドレスを保持する第一プログラムカウンタの値とプログラムカウンタ相対値の上位ビットとを加算し、前記第一プログラムカウンタが保持するパケットアドレスによって識別される命令パケットの中で、実行すべき命令を特定するパケット内アドレスを保持する第二プログラムカウンタの値とプログラムカウンタ相対値の下位ビットとを加算するプログラムカウンタ加算ステップをプロセッサに実行されるためのプログラムを記録した記録媒体。

【請求項 38】 前記プログラムカウンタ加算ステップは、さらに前記第一プログラムカウンタが保持するパケットアドレスと前記第二プログラムカウンタが保持するパケット内アドレスとを実行すべき命令の前記実行単位中の偏位だけ補正することを前記加算に先だって行なうことを特徴とする請求項 37 記載の記録媒体。

【請求項 39】 前記プログラムカウンタ加算ステップは、さらに前記第二プログラムカウンタの値と前記プログラムカウンタ相対値の前記下位ビットとの加算により生じた桁上がり数をも、前記第一プログラムカウンタの値とプログラムカウンタ相対値の上位ビットとの加算に加えることを特徴とする請求項 37 又は請求項 38 記載の記録媒体。

【請求項 40】 前記プログラムカウンタ加算ステップは、第二プログラムカウンタの値とプログラムカウンタ相対値の下位ビットとを加算することに代えて、前記プログラムカウンタ相対値の下位ビットを結果とすることを特徴とする請

求項 37 又は請求項 38 記載の記録媒体。

【請求項 41】 前記プログラムカウンタ加算ステップは、前記第一プログラムカウンタと前記第二プログラムカウンタとに加算結果を格納することを特徴とする請求項 37 から請求項 40 のいずれか 1 項に記載の記録媒体。

【請求項 42】 前記プログラムカウンタ加算ステップは、レジスタに加算結果を格納することを特徴とする請求項 37 又は請求項 38 又は請求項 39 又は請求項 40 記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プロセッサ及びそのプログラム開発環境に関する。特にプロセッサは、フェッチしたビット列に含まれる命令が必ずしもプロセッサが扱う情報単位であるバイト等の境界に位置していないことを実現するためのプログラムカウンタ及びプログラムカウンタの演算器を備えることを特徴とする。また、特にプログラム開発環境は、このようなプログラムカウンタの操作を行う分岐命令等を含む命令コードを開発できることを特徴とする。

【0002】

【従来の技術】

近年、プロセッサ応用商品のプログラム規模は増大している。そのため、大規模プログラムを効率よく実行できるプロセッサが要求されている。

【0003】

一方、従来の一般的なプロセッサの命令は 8 ビットを 1 バイトとするバイト単位にアドレッシングされている。

【0004】

従って、例えば 21 ビット長のプロセッサの命令は図 35 に示すような 3 バイト長すなわち 24 ビット長の命令として扱うことになる。

【0005】

【発明が解決しようとする課題】

しかしながら、上記従来技術では、図 35 に示した命令から明らかなように 1

命令につき 3 ビット未使用の領域が存在する。この未使用の領域によってプログラムサイズが増大するという問題がある。

【0006】

ここで仮に 1 バイトを 7 ビットとすると上記の未使用領域はなくなるが、8 ビットを 1 バイトとする汎用メモリが使えないという問題が生じる。

【0007】

そこで、本発明はかかる問題点に鑑みてなされたものであり、バイト単位にアドレッシングされない長さの命令を実行するプロセッサを提供することを第一の目的とする。さらにこのような長さの命令を含む命令コードを開発するプログラム開発環境を提供することを第二の目的とする。

【0008】

【課題を解決するための手段】

上記第一の目的を達成するために本発明のプロセッサは、命令あるいは分割された命令の一部を 2 の巾乗数ではない複数個含む命令パケットのうち、実行すべき命令またはその一部を含む命令パケットを識別するパケットアドレスを保持する第一プログラムカウンタ手段と、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中で、実行すべき命令を特定するパケット内アドレスを保持する第二プログラムカウンタ手段と、前記第二プログラムカウンタ手段が保持するパケット内アドレスを、次に実行すべき命令を特定するパケット内アドレスに更新し、さらに次に実行すべき命令またはその一部が、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットに含まれない場合は、前記第一プログラムカウンタ手段が保持するパケットアドレスを次に実行すべき命令を含む命令パケットのパケットアドレスに更新するプログラムカウンタ更新手段と、前記第一プログラムカウンタ手段が保持するパケットアドレスによって識別される命令パケットの中の命令のうち、前記第二プログラムカウンタ手段が保持するパケット内アドレスによって特定される命令を実行する実行手段とを備えることを特徴とする。

【0009】

また上記第二の目的を達成するために本発明のリンカは、複数のリロケータブ

ルコードを結合する結合手段と、前記複数のリロケータブルコードの中から2つの命令間のアドレスの差分に解決すべき再配置情報を検出する検出手段と、前記2つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算する下位減算手段と、前記2つの命令について、前記一方のアドレスの上位ビットから前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出した再配置情報を前記算出手段で算出した値で解決する解決手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とする。

【0010】

また上記第二の目的を達成するために本発明のアセンブラは、命令列をリロケータブルコードに変換する変換手段と、前記命令列の中から2つの命令間のアドレスの差分に解決すべきラベルを検出する検出手段と、前記2つの命令について一方のアドレスの下位ビットから他方のアドレスの前記下位ビットを減算する下位減算手段と、前記2つの命令について、前記一方のアドレスの上位ビットから前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出したラベルを前記算出手段で算出した値で解決する解決手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とする。

【0011】

また上記第二の目的を達成するために本発明の命令列最適化装置は、命令列の中から2つの命令間のアドレスの差分に解決すべきラベル及び命令のアドレスに解決すべきラベルを検出する検出手段と、前記命令列の各命令にアドレスを付与するアドレス付与手段と前記2つの命令について、前記アドレス付与手段が付与

した一方のアドレスの下位ビットから前記アドレス付与手段が付与した他方のアドレスの前記下位ビットを減算する下位減算手段と、前記 2 つの命令について、前記アドレス付与手段が付与した前記一方のアドレスの上位ビットから前記アドレス付与手段が付与した前記他方のアドレスの前記上位ビットを減算する上位減算手段と、前記上位減算手段が減算した結果が前記上位ビットであり、前記下位減算手段が減算した結果が前記下位ビットである値を算出する算出手段と、前記検出手段で検出したラベルを前記算出手段で算出した値に変換し、前記ラベルを有する命令を前記算出手段で算出した値に応じた命令に変換する変換手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とする。

【0012】

また上記第二の目的を達成するために本発明の逆アセンブラ装置は、オブジェクトコードの中に存在する命令ビットパターンを識別するためのアドレスの下位ビットと、前記命令ビットパターン中に指定される相対アドレス値の下位ビットとを加算する下位加算手段と、前記命令ビットパターンを識別するためのアドレスの上位ビットと、前記相対アドレス値の上位ビットとを加算する上位加算手段と、前記上位加算手段が加算した結果が上位ビットであり、前記下位加算手段が加算した結果が下位ビットである値を算出する算出手段と、前記値とラベル名とを対にして記憶する記憶手段と、前記算出手段が算出した結果に基づいて前記記憶手段を検索してラベル名を得る検索手段と、逆アセンブル時に、前記検索手段が検索したラベル名を表示する表示手段とを備え、アドレスの前記上位ビットによって、命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別し、アドレスの前記下位ビットによって、命令パケット中の命令を特定することを特徴とする。

【0013】

また上記第二の目的を達成するために本発明のコンパイラは、命令あるいは分割された命令の一部を 2 の巾乗ではない複数個含む命令パケットを識別するパケットアドレスを上位、前記命令パケットの中で実行すべき命令を特定するパケッ

ト内アドレスを下位とするプログラムカウンタ値と、プログラムカウンタ値の相対値とを、前記上位と前記下位に分離して加算する命令を生成するプログラムカウンタ値加算命令生成手段を備えることを特徴とする。

【0014】

また上記第二の目的を達成するために本発明のデバッガは、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットに含まれる対象命令のパケットアドレスを特定する命令特定手段と、前記命令特定手段が特定したパケットアドレスによって識別される命令を含む命令パケットのパケットアドレスを特定する命令パケット特定手段と、前記命令パケット特定手段が特定するパケットアドレスによって識別される命令パケットのビットパターンを検出するビットパターン検出手段と、前記ビットパターン検出手段が検出したビットパターンにおける前記対象命令に相当するビット列を、該ビット列と異なるビット列に更新するビットパターン更新手段とを備えることを特徴とする。

【0015】

また上記第二の目的を達成するために本発明の記録媒体は、上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2つの前記命令間の前記アドレスの差分に解決すべき再配置情報を、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で、解決する再配置情報解決ステップとをリンクに実行させるためのプログラムを記録することを特徴とする。

【0016】

また上記第二の目的を達成するために本発明の記録媒体は、上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2つの前記命令間の前記アドレ

スの差分に解決すべきラベルを、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で、解決するラベル解決ステップとをアセンブラに実行させるためのプログラムを記録することを特徴とする。

【0017】

また上記第二の目的を達成するために本発明の記録媒体は、上位ビットによって、命令あるいは分割された命令の一部を2の中乗ではない複数個含む命令パケットを識別し、下位ビットによって、命令パケット中の命令を特定するアドレスを命令に付加する命令アドレス決定ステップと、2つの前記命令間の前記アドレスの差分に解決すべきラベルを、2つの前記命令の一方の前記アドレスの前記下位ビットから他方の前記アドレスの前記下位ビットを減算した結果を前記下位ビットとし、一方の前記アドレスの前記上位ビットから他方の前記アドレスの前記上位ビットを減算した結果を前記上位ビットとする値で変換し、前記ラベルを有する命令を前記値に応じた命令に変換する第一命令変換ステップと、前記アドレスに解決すべきラベルを有する命令に対して、前記ラベルを前記命令アドレス決定ステップが付加したアドレスに変換し、このアドレスに応じた命令に変換する第二命令変換ステップとを命令列最適化装置に実行させるためのプログラムを記録することを特徴とする。

【0018】

また上記第二の目的を達成するために本発明の記録媒体は、命令あるいは分割された命令の一部を2の中乗数ではない複数個含む命令パケットのうち、実行すべき命令またはその一部を含む命令パケットを識別するパケットアドレスを保持する第一プログラムカウンタの値とプログラムカウンタ相対値の上位ビットとを加算し、前記第一プログラムカウンタが保持するパケットアドレスによって識別される命令パケットの中で、実行すべき命令を特定するパケット内アドレスを保持する第二プログラムカウンタの値とプログラムカウンタ相対値の下位ビットとを加算するプログラムカウンタ加算ステップをプロセッサに実行されるためのプ

ログラムを記録することを特徴とする。

【0019】

【発明の実施の形態】

以下、本発明に係るプロセッサ、コンパイラ、命令列最適化装置、アセンブラ、リンカ、デバッガ、逆アセンブラ装置の実施の形態について、図面を用いて詳細に説明する。

【0020】

(実施の形態1)

(プロセッサの命令フォーマットとアーキテクチャの概要)

まず、本発明に係るプロセッサが実行する命令の構造について説明する。

【0021】

図1(a)～(e)は本プロセッサの命令フォーマットを示す図である。本プロセッサの各命令は、21ビットの単位命令(以下、「ユニット」と呼ぶ)にて構成されており、ユニット1つで構成される21ビット命令とユニット2つで構成される42ビット命令の2種類の命令フォーマットが存在する(以下、「命令」とはこの21ビット命令あるいは42ビット命令のいずれかのことを指す)。各命令がいずれの長さの命令であるかは、1ビットのフォーマット情報101によって決定される。具体的には、フォーマット情報101が“0”の時はそのユニットが21ビット命令になり、フォーマット情報101が“1”の時はそのユニットとそれに後続するユニットを合わせて42ビット命令となる。

【0022】

また、各命令には1ビットの並列実行境界情報100を持たせてある。この情報は、この命令とそれに後続する命令との間に並列実行の境界が存在するか否かを示すものである。具体的には、並列実行境界情報100が“1”の時はその命令と後続命令の間に並列実行の境界が存在し、並列実行境界情報100が“0”の時には並列実行の境界が存在しないことになる。

【0023】

各命令の命令長からフォーマット情報101と並列実行境界情報100を除いた部分にてオペレーションを指定する。21ビット命令では19ビット、42ビ

ット命令では40ビットの長さを使用することができることになる。具体的には、“Op1”、“Op2”、“Op3”のフィールドでは、オペレーションの種類を表すオペコードを、“Rs”のフィールドでは、ソースオペランドとなるレジスタのレジスタ番号を、“Rd”のフィールドでは、デスティネーションオペランドとなるレジスタのレジスタ番号を指定する。また、“imm5”及び“imm32”のフィールドでは、それぞれ5ビットと32ビットの演算用定数オペランドを指定する。そして、“disp13”及び“disp31”のフィールドでは、それぞれ13ビットと31ビットの偏位（ディスプレースメント）を指定する。

【0024】

32ビットの定数などの長い定数を扱う転送命令や演算命令、大きなディスプレースメントを指定する分岐命令は42ビット命令で定義され、それらを除くほとんどの命令は21ビット命令で定義されている。なお、図3を見てわかるように、42ビット命令の構成要素である2つのユニットのうち、後ろの方のユニットすなわち2番目のユニットには、長い定数やディスプレースメントの一部のみが配置され、オペコードは配置されない。

【0025】

次に、本プロセッサのアーキテクチャの概要について説明する。

本プロセッサは、静的な並列スケジューリングを前提としたプロセッサであって、命令の供給と発行の概念は図2のようになる。命令の供給は、同図（a）に示すように毎サイクル64ビットの固定長の供給単位（以下、「パケット」と呼ぶ）でユニットを3個ずつ供給する。ユニット3個分の長さは63ビットであるが、残りの1ビットについては使用しない。そして、命令の実行は、同図（b）に示すように1サイクルで並列実行の境界までを同時実行する。つまり、各サイクルにおいて並列実行境界情報100が“1”である命令までの命令を並列実行することになる。供給されながら実行されずに残ったユニットは、命令バッファに蓄積され、次のサイクル以降で実行の対象となる。つまり、このアーキテクチャでは固定長のパケット単位で命令を供給しておき、静的に求めた情報を元に、各サイクルにおいて並列度に応じた適切な数のユニットの発行を行う。

【0026】

また、本プロセッサで用いられるいくつかの用語について説明する。「PC相対値」とは、2つの命令のアドレスの差分で表される値を意味する。「パケットアドレス」とはパケットを特定するアドレスを意味し、29ビットで表現する。

「パケット内アドレス」とは パケット内の位置を特定するアドレスを意味し、3ビットで表現する。「アドレス」とはパケットアドレスを上位ビット、パケット内アドレスを下位ビットとして表される値を意味し、メモリ内のユニットの位置を特定する。なお、これらの用語は、コンパイラ、最適化装置、アセンブラ、リンカ、デバッガ、逆アセンブラ装置においても、同様の意味で用いられる。

【0027】

(各装置の説明)

図3は、本発明に係るプロセッサ、コンパイラ、命令列最適化装置、アセンブラ、リンカから構成されるブロック図である。

【0028】

まずコンパイラ301は、高級言語で書かれたソースコード300の内容を解析し、アセンブラコード302を出力する。最適化装置303は、アセンブラコード302が有するラベルの中で、PC相対値に解決すべきラベルについて値を予測し（予測の方法については後述）、その値の大きさに応じた命令に変換し、最適化コード304として出力する。アセンブラ305は、最適化コード304を再配置可能なリロケータブルコード306に変換し出力する。PC相対値に解決可能なラベルについては値を算出し、その値をPC相対値としてラベルと置き換える。解決不可能なラベルについては再配置情報とする。リロケータブルコード306は機械語コードと再配置情報で構成する。リンカ307は、複数のリロケータブルコード306の結合を行い、オブジェクトコード308に変換し出力する。また、再配置情報をPC相対値あるいは命令のアドレスに解決する。プロセッサ309は、オブジェクトコード308を実行する。

【0029】

以下順に図3に示したプロセッサ309、リンカ307、アセンブラ305、最適化装置303、コンパイラ301についてその構成の詳細を記す。

【0030】

（プロセッサ）

図4は、図3に示したプロセッサ309の詳細を示すブロック図である。

【0031】

プロセッサ309は、最大3命令を並列実行可能なプロセッサであり、3つの演算が同時並列に実行可能な演算器401a～c、データ、アドレス等を格納する汎用レジスタ402、次に実行する命令の先頭アドレスを保持する29ビットの上位プログラムカウンタ403（以下、上位PC。値は16進数29ビットの29'h01234567等で表記）と3ビットの下位プログラムカウンタ404（以下、下位PC。値は2進数3ビットの3'b001等で表記。）、上位PCと下位PCの各種演算を行うPC演算器405、データを記憶するデータメモリ406と命令を記憶する命令メモリ407、さらに命令メモリ407からフェッチした命令を格納する命令バッファ408、命令バッファ408内の実行可能な命令をそれぞれデコードする命令デコーダ409a～cを備える。以下、上位

プログラムカウンタ 403 と下位プログラムカウンタ 404 を合わせてプログラムカウンタと称する。

【0032】

プロセッサ 309 の構成要素である上位 PC 403、下位 PC 404 は、次に実行可能な命令中の先頭命令のアドレスの上位 29 ビット、下位 3 ビットをそれぞれ保持する。

【0033】

プロセッサ 309 は、分岐命令以外の実行にあたって、後続する命令のアドレスにプログラムカウンタの値を更新する必要がある。この処理は、PC 演算器 405 に上位 PC 403、下位 PC 404 を入力し、デコードしたユニット数だけアドレスのインクリメント（このインクリメントの方法は後述する）を PC 演算器 405 が行い、その結果で上位 PC 403、下位 PC 404 を更新することで実現する。

【0034】

プロセッサ 309 は、分岐命令の実行にあたって、分岐先の命令のアドレスにプログラムカウンタの値を更新する必要がある。すなわちプログラムカウンタの値に分岐命令中に指定される PC 相対値を加算した結果にプログラムカウンタを更新する必要がある。この処理は、

- (1) 分岐命令をデコードした第 1 命令デコーダ 409 a あるいは第 2 命令デコーダ 409 b あるいは第 3 命令デコーダ 409 c から得られた PC 相対値
 - (2) 実行単位の先頭命令から分岐命令までに存在するユニットの数だけ、上位 PC 403、下位 PC 404 をインクリメントした値
- を PC 演算器 405 に入力し上記した PC 相対値とアドレス値の加算を行い、その結果で上位 PC 403、下位 PC 404 を更新し、また、プリフェッチカウンタ 410 へも演算結果を転送することで実現する。

【0035】

次に、本発明において特徴的である、命令のアドレス及び、インクリメントや PC 相対値との演算方法について詳細を記す。

【0036】

命令のアドレスは32ビットの値で定義される。その上位29ビットは命令パケットを特定するパケットアドレスである。下位3ビットは命令パケットに含まれる命令ユニットを特定するパケット内アドレスである。パケット内アドレスの値と対応するユニットを以下に示す。

【0037】

3' b000 パケット内の1番目のユニット

3' b010 パケット内の2番目のユニット

3' b100 パケット内の3番目のユニット

従って、アドレス値の下位3ビットは、3' b000、3' b010、3' b100のいずれかである。

【0038】

アドレス値のインクリメントは、パケット内アドレスが3' b000、3' b010の場合、下位3ビットの値に2を加算する。パケット内アドレスが3' b100の場合、パケットアドレスに桁上りを発生させて（すなわち上位29ビットの値に1を加算）、下位3ビットの値を3' b000とする。従ってパケット内アドレスのインクリメントは3' b000、3' b010、3' b100を巡回する3進演算となる。また、インクリメント値とパケット内アドレスの値の関係を図5に示す。例えば、インクリメント値が2で、インクリメント前のパケット内アドレスの値が3' b100であったとすると、インクリメント後のパケット内アドレスは3' b010となり、パケットアドレスに桁上りが1発生する。

【0039】

ここで、このプロセッサで用いられるアドレス計算方式の一つである桁上り方式について説明する。桁上り方式以外にも、分離方式、絶対位置指定方式、リニアアドレス方式があるが、それについては後述する。以下、コンパイラ、最適化装置、アセンブラ、リンカ、デバッガ、逆アセンブラ装置においても、対象とするプロセッサの方式に対応したアドレス計算方式が採用される。

【0040】

桁上り方式では、アドレスを上位29ビット、下位3ビットに分離して演算

する。ただし、上位ビットの演算の際、下位3ビットの演算で発生した桁上がり数または桁借り数を含めた演算が行われる。

【0041】

32ビットのPC相対値とアドレス値の加算について説明する。加算においては、上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの加算を図6(a)に示した加算表に従って行う。図6(a)はアドレスとPC相対値の下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、PC相対値の上位29ビットとアドレス値の上位29ビットと下位3ビットの加算によって発生した桁上がり数を加算する。これでPC相対値とアドレス値の加算が終了する。

【0042】

また、PC相対値を求める演算、すなわち2命令間のアドレス値の減算についても説明する。加算と同様に上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの減算を図6(b)に示した減算表に従って行う。図6(b)に示した減算表は2つのアドレスの下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、一方のアドレスの上位29ビットから他方の上位29ビットと下位3ビットの減算によって発生した桁借り数を減算する。これで2つのアドレス値の減算が終了する。

【0043】

(リンカ)

図7は、図3に示したリンカ307の構成及び関連する入出力データを示すブロック図である。

【0044】

リンカ307は、複数のリロケータブルコード701を結合し、各命令のアドレスを決定し、プロセッサが実行可能なオブジェクトコード714を出力する。その際、命令のアドレス、あるいは、PC相対値に解決すべきすべての再配置情報(ラベル)の値の算出を行うことが可能となる。

【0045】

コード結合手段702は、入力された複数のリロケータブルコード701の結

合を行い、すべての命令のアドレスを決定し、命令のアドレスに解決すべき再配置情報を決定されたアドレスにより解決して、結合コード703を出力する。

【0046】

次に、再配置情報検出手段704は、2つの命令間のアドレスの差分、すなわち、PC相対値に解決すべきラベルを検索し、再配置情報705として出力する。

【0047】

そして、再配置情報検出手段704で得られた再配置情報705を解決するため、以下、下位アドレス減算手段706、上位アドレス減算手段709、アドレス差算出手段711において、PC相対値の算出を行う。

【0048】

下位アドレス減算手段706は、一方の packets 内アドレスから他方の packets 内アドレスを減算し、桁借り数707と下位減算結果708の算出を行う。

【0049】

上位アドレス減算手段709は、一方の packets アドレスから他方の packets アドレスと下位アドレス減算手段706が算出した桁借り数707を減算し、上位減算結果710の算出を行う。

【0050】

そして、アドレス差算出手段711は、下位アドレス減算手段706が算出した下位減算結果708を下位ビット、上位アドレス減算手段709が算出した上位減算結果710を上位ビットとするアドレス差712を算出する。

【0051】

最後に、再配置情報解決手段713は、コード結合手段702が生成した結合コード中の再配置情報705をアドレス差算出手段711が算出したアドレス差712で解決し、オブジェクトコード714として生成する。

【0052】

(アセンブラ)

図8は、図3に示したアセンブラ305の構成及び関連する入出力データを示すブロック図である。

【0053】

アセンブラ305は、最適化装置303が生成した最適化コード801を再配置可能なリロケータブルコード814に変換する。その際、最適化コード801中の各パケットに29'h00000000を先頭とするパケットアドレス（以下、これを「局所パケットアドレス」と呼ぶ）を割り振ることにより、命令間のアドレスの差分を算出することが可能となるため、PC相対値に解決すべきラベルに関しては、その値の算出を行って、ラベルの解決を行う。

【0054】

機械語コード生成手段802は、最適化コード801をプロセッサ309が実行可能な機械語コード803への変換を行う。ただし、値が未解決なため、機械語コードへ変換できないラベルは、変換せずに機械語コード803中に保持する。

【0055】

次に、ラベル検出手段804は、2つの命令間のアドレスの差分、すなわち、PC相対値に解決すべきラベルを検索し、ラベル情報805として出力する。

【0056】

そして、ラベル検出手段804で得られたラベル情報805を解決するため、以下、下位アドレス減算手段806、上位アドレス減算手段809、アドレス差算出手段811において、PC相対値の算出を行う。

【0057】

下位アドレス減算手段806は、一方のアドレスの下位ビットから他方のアドレスの下位ビットを減算し、桁借り数807と下位減算結果808の算出を行う。

【0058】

上位アドレス減算手段809は、一方のアドレスの上位ビットから他方のアドレスの上位ビットと下位アドレス減算手段806が算出した桁借り数807を減算し、上位減算結果810の算出を行う。

【0059】

そして、アドレス差算出手段811は、下位アドレス減算手段806が算出し

た下位減算結果 808 を下位ビット、上位アドレス減算手段 809 が算出した上位減算結果 810 を上位ビットとするアドレス差 812 を算出する。

【0060】

最後に、ラベル情報解決手段 813 は、機械語コード生成手段 802 が生成した機械語コード 803 中のラベル情報 805 をアドレス差算出手段 811 が算出したアドレス差 812 で解決し、リロケータブルコード 814 として生成する。

【0061】

(最適化装置)

図 9 は、図 3 に示した最適化装置 303 の構成及び入出力データを示すブロック図である。

【0062】

最適化装置 303 は、コンパイラ 301 が生成したアセンブラコード 901 を入力とし、コードの最適化を行い、命令列を 3 ユニットを単位とするパケットに連結し、最適化コード 915 に変換する。その際、各命令のアドレスを予測して、命令のアドレスに解決すべきラベル、あるいは、PC 相対値に解決すべきラベルを含む命令が、そのラベルのサイズに応じて、21 ビット命令になるか、42 ビット命令になるかの決定を行うので、各命令の使用するユニット数が決まり、命令列を 3 ユニットを単位とするパケットに連結することが可能となる。

【0063】

コード最適化手段 902 においては、アセンブラコード 901 の最適化を行い、最適化処理コード 903 を生成する。コード最適化手段 902 の処理は、公知の最適化装置が有するコード最適化手段による処理と同様であるため、詳細な説明は省略する。

【0064】

次に、アドレス付与手段 904 においてコード最適化手段 902 により得られた最適化処理コード 903 の各命令のアドレスの予測を行い、各命令へ予測したアドレスの付与を行い（以下、このアドレスを「仮アドレス」と呼ぶ）、アドレス付与コード 916 を出力する。

【0065】

そして、ラベル検出手段 905 は、命令のアドレスに解決すべきラベル、と 2 つの命令間のアドレスの差分、すなわち、PC 相対値に解決すべきラベル、のラベル情報 906 の検出を行う。このとき、命令のアドレスに解決すべきラベルはその命令の仮アドレスの情報を、PC 相対値に解決すべきラベルは、2 つの命令の仮アドレスの情報を取り込む。

【0066】

そして、下位アドレス減算手段 907、上位アドレス減算手段 910、アドレス差算出手段 912 においては、ラベル情報 906 中の PC 相対値の算出を行っている。

【0067】

下位アドレス減算手段 907 は、前記 2 つの命令の一方の仮アドレスの下位ビットから他方の仮アドレスの下位ビットを減算し、桁借り数 908 と下位減算結果 909 の算出を行う。

【0068】

上位アドレス減算手段 910 においては、前記 2 つの命令の一方の仮アドレスの上位ビットから他方の仮アドレスの上位ビットと下位アドレス減算手段 907 が算出した桁借り数 908 を減算し、上位減算結果 911 の算出を行う。

【0069】

そして、アドレス差算出手段 912 において、下位アドレス減算手段 907 が算出した下位減算結果 909 を下位ビット、上位アドレス減算手段 910 が算出した上位減算結果 911 を上位ビットとするアドレス差 913 を算出する。

【0070】

ラベル情報解決手段 914 においては、コード最適化手段 902 が生成した最適化処理コード中のラベルを含む命令を、アドレス付与手段 904 が予測し付与したアドレスか、あるいは、アドレス差算出手段 912 が算出したアドレス差 913 の値から、その値の大きさに応じた命令に変換する。すなわち、前記付与アドレスあるいは前記アドレス差の値が 13 ビット以内で表現できれば 21 ビット命令とし、そうでなければ 42 ビット命令に変換する。そして最後に、変換後の命令列を 3 ユニットを単位とするパケットに連結し、最適化コード 915 として

生成する。

【0071】

以上、図3に示したプロセッサ309、リンカ307、アセンブラ305、最適化装置303についてその構成の詳細を記した。なお、コンパイラ301については、公知のコンパイラの構成と同様であるため、詳細な説明は省略する。

【0072】

次に、図3に示したプロセッサ309、リンカ307、アセンブラ305、最適化装置303に関して特徴的な動作を具体的な命令を用いて順に説明する。コンパイラ301の動作については、公知のコンパイラが行う動作と同様であるため、詳細な説明は省略する。

【0073】

(最適化装置の具体的な動作)

図10は、最適化装置303のコード最適化手段902において、コンパイラ301が出力したアセンブラコード302を最適化した最適化処理コード903の例の一部分を示している。図10のいくつかの命令について説明を行う。

【0074】

1000「L1:movr2, r1」: ラベルL1の位置を示す。レジスタr2からレジスタr1への転送を行う。

【0075】

1001「jsrf」: 関数呼出を行う。ラベルf(外部ラベル)への相対分岐を行い、ret命令により、このアドレスに再び戻る。fの大きさにより、21ビット命令か42ビット命令かが決定する。

【0076】

1002「addr0, r4」: レジスタr0とレジスタr4の加算を行い、結果をr4へ格納する。

【0077】

1003「andr1, r3」: レジスタr1とr3の論理和を取り、結果をr3へ格納する。

【0078】

1005「ld(r2), r0」: レジスタr2が示すアドレスに格納されているデータをレジスタr0へ転送する。

【0079】

1006「braL1」: ラベルL1（局所ラベル）への相対分岐を行う。

なお、図10において、命令1007に後続する命令が省略してあるが、この後続する命令中には、ラベルfの位置する命令は存在しないこととする。

【0080】

アドレス付与手段904においては、コード最適化手段902が出力した図10の最適化処理コード903の各命令へアドレスの付与を行い、図11に示したアドレス付与コード916に変換される。この場合、32'h00000800から始まるアドレス（仮アドレス）が割り当てられている。

【0081】

ラベル検出手段905においては、図11のアドレス付与コード916中からラベルの検索を行い、命令のアドレスに解決すべきラベルとして命令1104のL2、PC相対値に解決すべきラベルとして命令1106のL1を検出し、図12のラベル情報906として出力する。図12は、ラベルL2については、L2を含む命令「movL2, r2」と解決すべきアドレスの情報を取り込み、ラベルL1については、L1を含む命令「braL1」とその差分に解決すべき2つのアドレスの情報を取り込んでいる。なお、命令1101のラベルfは外部ラベルであるため、最適化の対象外とする。

【0082】

下位アドレス減算手段907においては、PC相対値であるL1が示す値の下位ビットを計算する。分岐先命令1100の仮アドレス32'h00000800の下位3ビット3'b000から分岐命令1106の仮アドレス32'h00000812の下位3ビット3'b010を減算し、桁借り数908が1となり、下位減算結果909が3'b100となる。

【0083】

上位アドレス減算手段910においては、PC相対値であるL1が示す値の上位ビットを計算する。分岐先命令1100の仮アドレスの上位29ビット29'

h00000100から分岐命令1106の仮アドレスの上位29ビット29'h00000102と桁借り数908である1を減算し、上位減算結果911が29'h1ffffffffd(10進数では-3を示す。以下、負数は2の補数で表す。)となる。

【0084】

アドレス差算出手段912においては、3'b100を下位ビット、29'h1ffffffffdを上位ビットして、アドレス差913である32'hfffffffffecを算出する。

【0085】

ラベル情報解決手段914においては、ラベル情報906のラベルL1とラベルL2の値を解決して、そのラベルを含む命令1104と命令1106が21ビット命令であるか、42ビット命令であるかを決定する。まず、ラベルL2が示す値は32'h12345678であり、L2を含む命令1104を21ビット命令で表せないため、42ビット命令とする。また、ラベルL1が示す値は32'hfffffffffecであり、13ビットで表すことが可能であり、L1を含む命令1106は21ビット命令とする。さらに、命令列を3ユニットを単位とするパケットに連結する。その結果、図13に示した最適化コード915に変換される。

【0086】

図13は、一つのパケットを構成する命令が一行に記述され、記号||によりパケット中の命令は区切られている。また、42ビット命令は、()にくくられた命令を後続させることにより、その命令が2命令ユニット分の領域を使用することを示している。

【0087】

このように、桁上がり方式によるアドレス演算を行って予測することにより、桁上がり方式に対応したプロセッサの最適化装置を実現することが可能となる。

【0088】

なお、アドレス付与手段904において付与された仮アドレス、アドレス差算出手段912において算出されたPC相対値は、すべてラベルを含む命令のサイ

ズを確定させるために予測された値であって、実際の値とは異なる場合がある。そのため、この最適化装置で扱われたアドレスやPC相対値は、以降の処理では用いられない。

【0089】

(アセンブラの具体的動作)

最適化装置303が出力した図13の最適化コードをアセンブラ305に入力した場合を例に説明する。

【0090】

まず、機械語コード生成手段802において、パケット毎に、命令1300～1303のアセンブラコードからプロセッサ309に対応した機械語コード803を生成する。また、29'h00000000から始まるパケットアドレス(局所パケットアドレス)を割り当てる。ただし、パケット1300の命令「jsrf」中のラベルf、パケット1301の命令「movL2, r2」中のラベルL2、パケット1302の命令「braL1」中のラベルL1は、値が未解決なため、変換は行わない。生成された機械語コード803を図14に示す。なお、実際の機械語コードは、0と1だけのバイナリ形式で表されているが、図14においては可読性を高めるため、ニーモニック形式で表されている。また、このとき、各命令の並列実行境界情報100、ビットフォーマット情報101が明らかになっているが、図14においては特に示さない。

【0091】

次に、ラベル検出手段804において、図14に示した機械語コード803中の未解決なラベルのうち、PC相対値に解決可能なラベルである命令1409中のラベルL1を検出し、図15のラベル情報805として出力する。図15では、ラベルL1の値の解決に必要なアドレスを2つ取り込んでいる。

【0092】

下位アドレス減算手段806においては、L1の位置する命令1401のパケット内アドレス3'b000からL1を含む命令1409のパケット内アドレス3'b010を減算し、桁借り数807が1となり、下位減算結果808が3'b100となる。

【0093】

上位アドレス減算手段809においては、L1の位置するパケット1400の局所パケットアドレス29'h00000000から、L1を含む命令1409の局所パケットアドレス29'h00000002と下位アドレス減算手段806において発生した桁借り数807である1を減算する。その結果を29ビットで得ると、上位減算結果810は29'h1ffffffdとなる。

【0094】

アドレス差算出手段811で、上位減算結果810である29'h1ffffffdを上位29ビットとし、下位減算結果808である3'b100を下位3ビットとした32ビットの値32'hffffffecを算出する。

【0095】

ラベル情報解決手段813においては、未解決であったL1に32'hffffffecの下位13ビットの値13'h1fecを埋め込むことにより解決する。

【0096】

このようにして生成されるリロケータブルコード814の出力を図16に示す。図16では、機械語コード803の出力時にすでに明らかになっていた各命令の並列実行境界情報100とビットフォーマット情報101が示され、パケット中の1ビットが未使用ビットになっていることも示されている。

【0097】

このように、桁上がり方式によるアドレス演算を行ってPC相対値を求めることにより、桁上がり方式に対応したプロセッサのアセンブラを実現することが可能となる。

【0098】

(リンカの具体的動作)

アセンブラ305が出力した図16のリロケータブルコード814と、別途生成された図17のリロケータブルコードをリンカ307に入力した場合を例に説明する。この例では図17に示したリロケータブルコードの後ろに図16に示したリロケータブルコードを結合し、先頭のパケットアドレスは29'h0000

0000とする。

【0099】

まずコード結合手段702において、2つのリロケータブルコード701を結合する。すなわち、図17に示したコードの先頭の packets アドレスを29'h00000000、図16に示したコードの先頭の packets アドレスを29'h00000001と決定し、すべての命令のアドレスを決定する(図18)。さらに、命令1810「movl2, r2」のラベルL2のアドレスは、packets 1815の先頭のアドレスにより32'h12345680と決定しているので、これを解決した結合コード703を図19に示す。

【0100】

次に、再配置情報検出手段704においては、PC相対値に解決すべきラベルを検索し、命令1906のラベルfを得て、図20に示す再配置情報705を出力する。

【0101】

下位アドレス減算手段706においては、fの位置する命令1901のアドレスの下位3ビット3'b000からPC相対値fを含む命令1906のアドレスの下位3ビット3'b010を減算し、桁借り数707が1となり、下位減算結果708が3'b100となる。

【0102】

上位アドレス減算手段709においては、fの位置する命令1901のアドレスの上位29ビット29'h00000000から、PC相対値fを含む命令1906のアドレスの上位29ビット29'h00000001と下位アドレス減算手段において発生した桁借り数707である1を減算し、上位減算結果710が29'h1fffffffeとなる。

【0103】

アドレス差算出手段711においては、上位減算結果710である29'h1fffffffeを上位ビットとし、下位減算結果708である3'b100を下位ビットした32ビットのアドレス差32'hffffff4を算出する。

【0104】

再配置情報解決手段 713 においては、未解決であった PC 相対値の f を 32' h f f f f f f f 4 と決定し、命令 1906 には 13 ビットで表した値 13' h 1 f f 4 を埋め込む。出力されたオブジェクトコードを図 21 に示す。

【0105】

このように、桁上がり方式によるアドレス演算を行って PC 相対値を求めることにより、桁上がり方式に対応したプロセッサのリンカを実現することが可能となる。

【0106】

(プロセッサの具体的動作)

図 21 に示したリンカ 307 が出力したオブジェクトコードが図 4 に示すプロセッサの命令メモリ 407 に蓄えられているときの例を説明する。プリフェッチカウンタは 29' h 00000001、上位 PC は 29' h 00000001、下位 PC は 3' b 100 とする。

【0107】

まず、命令メモリ 407 に蓄えられた命令列のうち、プリフェッチカウンタ 410 により示されるパケット 2104 をプリフェッチして、命令バッファ 408 に入れられる。そのとき、プリフェッチカウンタ 410 はインクリメントされ、29' h 00000002 となる。以下、同様にして、プリフェッチカウンタ 410 に示されたパケットが命令メモリ 407 から命令バッファ 408 へプリフェッチされていく。

【0108】

そして、上位 PC 403 によりパケット 2104 が指定され、さらに下位 PC 404 によりパケット 2104 中の命令 2107 が指定されているため、次に実行可能な命令の先頭の命令が命令 2107 であることがわかる。命令 2107 を含むパケットが命令バッファ 408 にプリフェッチされたら、命令バッファ 408 中の実行可能な命令の先頭の命令 2107 から並列実行境界情報 100 を調べる。その結果、命令 2107、命令 2109、命令 2110 が同時並列実行可能であることがわかる。次に、それぞれの命令が、第 1 命令デコーダ 409a、第 2 命令デコーダに 409b、第 3 命令デコーダ 409c に送られる。そして、こ

のとき4つの命令ユニットがデコードされるので、アドレスはPC演算器405において4つインクリメントされ、下位PC404は3' b000となり、上位PC403は桁上がりで2つ起こり29' h00000003となる。

【0109】

そして、第1命令デコーダ409aにおいて、命令「add r0, r4」がデコードされ、出された制御信号によりレジスタr0とレジスタr4の値を入力として第1演算器401aにおいて加算が行われ、その結果はレジスタr4へ格納される。第2命令デコーダ409bにおいては、命令「and r1, r3」がデコードされ、出された制御信号により、レジスタr1とレジスタr3の値を入力として第2演算器401bにおいて論理演算が行われ、その結果はレジスタr3へ代入される。また、第3命令デコーダ409cにおいては、命令「mov 32' h12345680, r2」がデコードされ、出された制御信号により、第3命令デコーダ409cから送られた即値の32' h12345680が、レジスタr2に転送される。

【0110】

次に、上位PC403は29' h00000003で、下位PC404は3' b000であるため、次の実行単位の先頭命令は命令2112であり、並列実行境界情報100を調べると命令2112は“0”、命令2113は“1”になっているので、次に同時並列実行可能な命令は命令2112、2113であることがわかる。そこで、それぞれの命令は、第1命令デコーダ409a、第2命令デコーダ409bへ送られる。

【0111】

そして、第1命令デコーダ409aにおいて、命令「ld (r2), r0」がデコードされ、レジスタr2の値をアドレスとするデータメモリ中のデータを取り込み、レジスタr0へ格納する。また、第2命令デコーダ409bにおいて、命令「bra 13' h1fec」をデコードした結果、命令2113が分岐命令であることが判明したため、アドレスを分岐先命令のアドレスに更新する必要がある。その処理の詳細を以下に示す。

【0112】

まず、上位PC 403、下位PC 404の値を、実行単位の先頭命令から分岐命令までに存在するユニット数1だけPC演算器でインクリメントを行い、上位PCが29'h00000003となり、下位PCが3'b010となる。そして、得られた上位PC、下位PCと第2命令デコーダ409bから得られたPC相対値13'h1f e cをPC演算器405に入力し、アドレスとPC相対値の加算を行う。PC相対値は、32ビットに符号拡張した値32'h f f f f f f e cを用いる。加算は、上位29ビットと下位3ビットに分離して行われ、まず下位PC 3'b010とPC相対値の下位3ビット3'b100の加算を行う。その結果、桁上がり数1と下位演算結果3'b000が得られ、下位演算結果は下位PC 404に転送される。次に、上位PC 29'h00000003、PC相対値の上位29ビット29'h1f f f f f f d、桁上がり数1の3数を加算し、上位演算結果29'h00000001が得られ、上位演算結果は上位PC 403とプリフェッチカウンタ410に転送される。

【0113】

以上の処理により、プリフェッチカウンタ410は29'h00000001となり、次にプリフェッチするパケットは、パケット2104となる。また、上位PC 403は29'h00000001、下位PC 404は3'b000となり、次の実行単位の先頭命令は命令2105となる。同様にして、命令の実行が行われていくが、以下の命令については省略する。

【0114】

このように、桁上がり方式によるアドレス演算を行うことによって、プログラムカウンタとPC相対値の加算の上位と下位の自然な連続性を表現でき、PC相対分岐を実現できる。

【0115】

(実施の形態2)

図28は本発明の第2の実施の形態におけるプロセッサの構成図である。このプロセッサは実施の形態1で示した図4に示すプロセッサに第2PC演算器2801を付加した構成である。第2PC演算器2801は図4に示したPC演算器405と同様の演算を行う。

【0116】

図30は第2PC演算器2801を使用する命令の二モニックとオペレーションの対応を示す図であり、(a)はPC相対値dispとレジスタの値を加算し、レジスタに結果を格納するPC加算命令、(b)はレジスタの値からPC相対値dispを減算し、レジスタに結果を格納するPC減算命令である。PC加算命令とPC減算命令は第2PC演算器2801を使用するので、実施の形態1で示したPCの加減算方法に従って演算を行う。

【0117】

図29は本発明の第2の実施の形態におけるコンパイラの構成図である。

ソースコード2901は高級言語で記述されたプログラムである。中間コード変換部2902はソースコード2901をコンパイラの内部表現である中間コード2903に変換する。中間コード変換部2902は公知の技術であるため、詳細な説明を省略する。PC値加算命令変換部2904は中間コード2903の中でPCと変数を加算する中間コードを図22に示した(a)PC加算命令に変換する。命令変換部2905はその他の中間コードを対応するアセンブラ命令に変換する。命令変換部2905は公知の技術であるため、詳細な説明を省略する。アセンブラコード2906はコンパイラが出力する。

【0118】

次に図29に示したコンパイラの動作を、例を用いて説明する。

図31はC言語で記述されたソースプログラムである。図中、関数g1, g2, g3, g4は宣言され、関数fはint型の変数iを受け取る関数として定義されている。関数fは、iの値が1ならばポインタfpに関数g1のアドレスを代入し、iの値が2ならばポインタfpに関数g2のアドレスを代入し、iの値が3ならばポインタfpに関数g3のアドレスを代入し、iが以上の値でなければポインタfpに関数g4のアドレスを代入する。最後にfpの指す関数を呼び出す。

【0119】

図32は図31に示したソースプログラムを中間コード変換部2902が変換した中間コードである。図中3201は関数の先頭のラベルfを有し、一時変数

t m p に P C の値を代入する中間コードである。図中 3 2 0 2 は変数 i が 1 と等しくないかを判定する中間コードである。図中 3 2 0 3 は前の中間コードの判定が真の場合ラベル L 1 へ分岐する中間コードである。図中 3 2 0 4 は関数 g 1 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値と一時変数 t m p を加算しその結果を変数 f p に代入する中間コードである。図中 3 2 0 5 はラベル L へ分岐する中間コードである。図中 3 2 0 6 はラベル L 1 を有し、変数 i が 2 と等しくないかを判定する中間コードである。図中 3 2 0 7 は前の中間コードの判定が真の場合ラベル L 2 へ分岐する中間コードである。図中 3 2 0 8 は関数 g 2 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値と一時変数 t m p を加算しその結果を変数 f p に代入する中間コードである。図中 3 2 0 9 はラベル L へ分岐する中間コードである。図中 3 2 1 0 はラベル L 2 を有し、変数 i が 3 と等しくないかを判定する中間コードである。図中 3 2 1 1 は前の中間コードの判定が真の場合ラベル L 3 へ分岐する中間コードである。図中 3 2 1 2 は関数 g 3 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値と一時変数 t m p を加算しその結果を変数 f p に代入する中間コードである。図中 3 2 1 3 はラベル L へ分岐する中間コードである。図中 3 2 1 4 はラベル L 3 を有し、関数 g 4 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値と一時変数 t m p を加算しその結果を変数 f p に代入する中間コードである。図中 3 2 1 5 はラベル L を有し、変数 f p が指す関数を呼び出す中間コードである。図 3 2 に示した中間コードは関数 g 1, g 2, g 3, g 4 の絶対アドレスを変数 f p に代入するのではなく、関数 f の先頭アドレスとそこから関数 g 1, g 2, g 3, g 4 への P C 相対値とを加算し変数 f p に代入する。この変換は上記したように公知の技術であるため、詳細な説明を省略する。

【0120】

図 3 3 は図 3 2 に示した中間コードを命令変換部 2 9 0 5 と P C 値加算命令変換部 2 9 0 4 が変換したアセンブラコードである。図中 3 3 0 1 は関数の先頭のラベル f を有し、レジスタ r 1 に P C を転送する命令である。図中 3 3 0 2 は 1 とレジスタ r 0 が等しくないかを判定する命令である。図中 3 3 0 3 は前に実行した命令の判定が真の場合ラベル L 1 へ分岐する命令である。図中 3 3 0 4 は関数

g 1 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値とレジスタ r 1 を加算しその結果をレジスタ r 1 に転送する命令である。図中 3 3 0 5 はラベル L へ分岐する命令である。図中 3 3 0 6 はラベル L 1 を有し、2 とレジスタ r 0 が等しくないか判定する命令である。図中 3 3 0 7 は前に実行した命令の判定が真の場合ラベル L 2 へ分岐する命令である。図中 3 3 0 8 は関数 g 2 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値とレジスタ r 1 を加算しその結果をレジスタ r 1 に転送する命令である。図中 3 3 0 9 はラベル L へ分岐する命令である。図中 3 3 1 0 はラベル L 2 を有し、3 とレジスタ r 0 が等しくないか判定する命令である。図中 3 3 1 1 は前に実行した命令の判定が真の場合ラベル L 3 へ分岐する命令である。図中 3 3 1 2 は関数 g 3 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値とレジスタ r 1 を加算しその結果をレジスタ r 1 に転送する命令である。図中 3 3 1 3 はラベル L へ分岐する命令である。図中 3 3 1 4 はラベル L 3 を有し、関数 g 4 の先頭アドレスから関数 f の先頭のアドレスを減算した相対値とレジスタ r 1 を加算しその結果をレジスタ r 1 に転送する命令である。図中 3 3 1 5 はラベル L を有し、r 1 が指す関数を呼び出す命令である。図中 3 3 1 6 は関数を終了する命令である。

【0121】

PC 値加算命令変換部 2 9 0 4 は図 3 2 中 3 2 0 4、3 2 0 8、3 2 1 2、3 2 1 4 の各中間コードを図中 3 3 0 4、3 3 0 8、3 3 1 2、3 3 1 4 の各命令に変換する。この変換は 3 2 0 4 の中間コードが有する被演算子 t m p が 3 2 0 1 で PC の値を転送されているため、演算子 + は PC と PC 相対値の加算であることを検出し、第 2 PC 演算器 2 8 0 1 を使用して加算を行う a d d p c に変換する。3 2 0 8、3 2 1 2、3 2 1 4 の各中間コードに対しても同様である。図中 3 3 0 4、3 3 0 8、3 3 1 2、3 3 1 4 以外の命令は命令変換部 2 9 0 5 によって変換される。この変換は上記したように公知の技術であるため、詳細な説明を省略する。

【0122】

図 3 3 に示したアセンブラコードは関数 g_1 , g_2 , g_3 , g_4 の絶対アドレスをレジスタ r_1 に転送する命令ではなく、関数 f の先頭アドレスとそこから関数 g_1 , g_2 , g_3 , g_4 への PC 相対値とを加算しレジスタ r_1 に転送する命令である。一般に絶対アドレスよりも PC 相対値の方が占有するメモリ領域は少ない。従って本実施の形態のコンパイラを使用することで、コードサイズを縮小することができる。また、プログラムを配置するアドレスが実行時に決定する（一般に PIC コードと呼ばれる）場合、分岐命令は図 3 3 に示したような PC 相対値を用いて分岐しなければならない、本実施の形態のコンパイラが必須である。本実施の形態のコンパイラが出力した命令を実行するには本実施の形態のプロセッサが必須である。

【0123】

（実施の形態 3）

図 3 4 は本発明の一実施の形態におけるデバッガ及び逆アセンブラの構成図である。命令メモリ 3401 はデバッグ及び逆アセンブルの対象となる命令コードを保持する。図に示したようにパケットアドレスとパケット内アドレスによって各命令のアドレスが決まる。各命令のアドレスはパケットアドレスを上位とし、パケット内アドレスを下位とする値である。図中では例として図 2 1 に示した命令コードが格納されている。図 2 1 に示した命令コードは実施の形態 1 で説明したので、ここでは詳細な説明を省略する。プロセッサ 3402 は図 4 に示すプロセッサから構成され、命令メモリ 3401 が保持する命令コードを実行する。ラベルテーブル 3403 はラベル名と命令アドレスの対応を保持するテーブルである。図中では例としてアドレス 32' h00000000 とラベル名 f 、アドレス 32' h00000008 とラベル名 L_1 、アドレス 32' h12345680 とラベル名 L_2 が対応している。命令バッファ 3404 は命令パケットを一つを保持する。64 ビットを保持し、最左ビットを 0 ビット目、最右ビットを 63 ビット目とする。表示部 3405 は命令コードを逆アセンブルした結果を表示する。PC 加算器 3406 はプログラムカウンタ値とプログラムカウンタ相対値を加算する演算器である。その加算方法は図 4 に示した PC 演算器 405 と同様であり、実施の形態 1 において詳細を説明したので省略する。制御部 3407 はデ

バッガ及び逆アセンブラを使用するオペレータが指示する操作内容に応じて他の構成要素を制御する。受け付ける入力以下の2つである。

- (a) 逆アセンブルのコマンドと逆アセンブルする命令のアドレス。
- (b) 命令を新しい命令に置き換えるコマンドと置き換える命令のアドレスと新しい命令の内容。

【0124】

本発明と関連のない他の機能は公知の技術であるので省略する。

次に(a)の入力を受け付けた場合の逆アセンブラの動作を以下に示す。

(1) 制御部3407に入力されたアドレスの上位の値からパケットアドレスを検出し、そのアドレスのパケットを命令メモリ3401から取り出し、命令バッファ3404に格納する。

(2) 制御部3407に入力されたアドレスの下位の値からパケット内アドレスを検出し、そのアドレスの命令パケットを命令バッファ3404から取り出す。

(3) 取り出した命令が分岐命令であれば、命令中に指定されるPC相対値と制御部3407に入力された命令のアドレスをPC加算器3406に入力し加算を行う。

(4) PC加算器3406が加算した値をアドレスとするラベル名を、ラベルテーブル3403中から検索する。

(5) 取り出した命令が分岐命令であれば、分岐命令のアセンブラ名と検索したラベル名を表示部3405によって表示する。他の命令であれば、アセンブラ名を表示部3405によって表示する。

【0125】

以上に示した動作の例を命令メモリ3401に示した命令列の例に即して以下に示す。

(a) 制御部3407にアドレス32'h0000001bを入力する。

(b) パケットアドレス3'b010に相当する命令パケットを命令バッファ3404に格納する。命令バッファ3404の内容はaddr0, r1||mov r0, r1||bcc0b100である。

(c) 命令バッファ3404の中でパケット内アドレス100に相当する命令は

b c c 0 b 1 0 0であり、分岐命令である。

(d) b r a 1 3' h 1 f e cの1 3' h 1 f e cと制御部3407に入力されたアドレス3 2' h 0 0 0 0 0 0 1 bを実施の形態1で説明した桁上がり方式に従ってPC加算器3406が加算する。下位3ビットを加算し、その桁上がりを含めて上位29ビットを加算し、結果は3 2' h 0 0 0 0 0 0 0 8になる。

(e) ラベルテーブル3403を検索して、アドレス3 2' h 0 0 0 0 0 0 0 8に対応するラベル名がL 1であることがわかる。

(f) 分岐命令のアセンブラ名であるb r aと検索したラベル名であるL 1を表示部3405によって表示する。

【0126】

次に(b)の入力を受け付けた場合のデバッガの動作を以下に示す。

(1) 制御部3407に入力されたアドレスの上位の値からパケットアドレスを検出し、そのアドレスのパケットを命令メモリ3401から取りだし、命令バッファ3404に格納する。

(2) 制御部3407に入力されたアドレスの下位の値からパケット内アドレスを検出する。

(3) パケット内アドレスが3' b 0 0 0 0ならば0ビット目から20ビット目を、3' b 0 1 0 0ならば21ビット目から41ビット目を、3' b 1 0 0 0ならば42ビット目から62ビット目を制御部3407に入力された命令のビットパターンに変更する。

(4) 制御部3407に入力されたアドレスの上位の値からパケットアドレスを検出し、そのアドレスのパケットを命令バッファ3404の内容に更新する。

【0127】

以上に示した動作の例を命令メモリ3401に示した命令列の例に即して以下に示す。

(a) 制御部3407にアドレス3 2' h 0 0 0 0 0 0 1 bとs u b r 0, r 1を入力する。s u b r 0, r 1は減算命令である。

(b) 命令メモリ3401からパケットアドレス2 9' h 0 0 0 0 0 0 0 3に相当する命令パケットを命令バッファ3404に格納する。命令バッファ3404

の内容は $ld(r2), r0 || bra13' h1fec || addr2, r3$ である。

(c) パケット内アドレスが $3' b010$ であるので、命令バッファ3404の21ビットから41ビットを $subr0, r1$ のビットパターンに変更する。

(d) 命令メモリ3401のパケットアドレス $29' h00000003$ に相当する命令パケットを命令バッファ3404の内容に更新する。

【0128】

本実施の形態では実施の形態1に記したプロセッサが実行する命令コードを逆アセンブルし、分岐命令である場合プログラムカウンタ相対値が数値でなくラベル名で表示できる。また、命令コードの命令を書き換えることもできる。

【0129】

以上、本発明に係る命令列最適化装置、アセンブラ、リンカ、プロセッサに関する実施の形態を説明したが、本発明はこれら実施の形態に限られないことは勿論である。即ち、

(1) 実施の形態1において、命令パケットのサイズは実行時に可変であってもよい。ただしプログラムカウンタのインクリメント時、上位プログラムカウンタ403は次に実行すべき命令の存在する命令パケットのアドレスに更新する必要があり、これは常に1加算するとは限らない。

(2) 実施の形態1において、第1命令デコーダ409a、第2命令デコーダ409b、第3命令デコーダ409cのうち、一つだけ有するプロセッサでもよい。

(3) 実施の形態1において、分岐命令のPC相対値は実行単位の先頭アドレスから分岐先アドレスへの差分値でもよい。すなわち、アセンブラ305において、図14の命令1409中のラベルL1の値は、分岐先アドレス $32' h00000000$ から命令1409の実行単位の先頭である命令1408のアドレス $32' h00000012$ の減算値となる。リンカ307においても、同様にPC相対値を求める。その結果、リンカ307が出力するオブジェクトコードは図22となる。図22では、図21に比べ命令2206、命令2213のPC相対値が異なっている。そして、プロセッサ309においては、分岐命令の実行時に、

上位PC403、下位PC404の値を実行単位の先頭命令から分岐命令までに存在するユニット数だけインクリメントするという処理が省かれ、分岐命令をデコードした第1命令デコーダ409aまたは第2命令デコーダ409bまたは第3命令デコーダ409cから得られたPC相対値と上位PC403、下位PC404の値の加算をPC演算器405で行う。

(4) 実施の形態1において、最適化装置303の機能をコンパイラ301に取り込んで行うこともできる。

(5) 実施の形態1において、アセンブラコード302、最適化コード304、リロケータブルコード306、オブジェクトコード308はマスクROM、フラッシュメモリ等の半導体集積メモリや、フロッピーディスク、ハードディスク等の磁気記録媒体や、CD-ROM、DVD等の光ディスクに記録することもできる。

(6) 実施の形態1において、アドレスの計算方式として、桁上がり方式に替えて、以下に示す分離方式、絶対位置指定方式、リニアアドレス方式のいずれかでも代用でき、それによってプロセッサ、リンカ、アセンブラ、最適化装置も実現することもできる。

【0130】

(分離方式)

分離方式は、アドレスを上位29ビット、下位3ビットに分離して演算する。ただし、桁上がり方式の場合と異なり、上位ビットの演算の際、下位3ビットの演算で発生した桁上がりあるいは桁借りは無視する。

【0131】

32ビットのPC相対値とアドレス値の加算について説明する。加算においては、上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの加算を図23(a)に示した加算表に従って行う。桁上がり方式と異なる点は、桁上りを無視する点である。図23(a)はアドレスとPC相対値の下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、PC相対値の上位29ビットとアドレス値の上位29ビットとの加算をする。これでPC相対値とアドレス値の加算が終了する。

【0132】

また、PC相対値を求める演算、すなわち2命令間のアドレス値の減算についても説明する。加算と同様に上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの減算を図23(b)に示した減算表に従って行う。桁上がり方式と異なる点は、桁借りを無視する点である。図23(b)に示した減算表は2つのアドレスの下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、一方のアドレスの上位29ビットから他方の上位29ビットを減算する。これで2つのアドレス値の減算が終了する。

【0133】

この方式の場合、前述のリンカ307を例にとると図24のオブジェクトコードが出力される。命令2406と命令2413のPC相対値が図21と異なっている。命令2406のPC相対値の算出法について説明する。下位3ビットについては、命令2401の packets 内アドレス3' b000から命令2406の packets 内アドレス3' b010を減算表図23(b)に従って減算した結果、3' b100となり、桁借りは無視される。上位29ビットについては、命令2401の packets アドレス29' h000000000から命令2406の packets アドレス29' h000000001を減算して、結果は29' h1ffffffffffとなる。よって、その2つの演算結果を結合したPC相対値は、13ビットで表現可能であり、13' h1ffffcとなる。

【0134】

この方式によると、プログラムカウンタとPC相対値の加算を単純なハードウェアで実現できる。

【0135】

(絶対位置指定方式)

絶対位置指定方式についても、アドレスを上位29ビット、下位3ビットに分離して演算うが、桁上がり方式、分離方式と異なる点は、PC相対値の下位3ビットとして、分岐先の packets 内アドレスの値をそのまま用いる点である。当然、上位ビットの演算の際に、下位ビットからの桁上がり、桁借りは発生しない。

【0136】

32ビットのPC相対値とアドレス値の加算について説明する。加算においては、上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの加算を図25(a)に示した加算表に従って行う。桁上がり方式、分離方式と異なる点は、PC相対値の下位3ビットがそのまま加算結果の下位3ビットとなることである。図25(a)はアドレスとPC相対値の下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、PC相対値の上位29ビットとアドレス値の上位29ビットとの加算をする。これでPC相対値とアドレス値の加算が終了する。

【0137】

また、PC相対値を求める演算、すなわち2命令間のアドレス値の減算についても説明する。加算と同様に上位29ビットと下位3ビットに分離して計算する。まず、下位3ビットの減算を図25(b)に示した減算表に従って行う。桁上がり方式、分離方式と異なる点は、一方のアドレスの下位3ビットがそのまま減算結果、すなわちPC相対値の下位3ビットとなることである。図25(b)に示した減算表は2つのアドレスの下位3ビットがとりうる値すべてについて演算結果を記した表である。その後、一方のアドレスの上位29ビットから他方の上位29ビットを減算する。これで2つのアドレス値の減算が終了する。

【0138】

この方式の場合、前述のリンカ307を例にとると図26のオブジェクトコードが出力される。命令2606と命令2613のPC相対値が図21と異なっている。命令2606のPC相対値の算出法について説明する。下位3ビットについては、命令2601の packets 内アドレス3' b000から命令2606の packets 内アドレス3' b010を減算表の図23(b)に従って減算した結果、2601の packets 内アドレス3' b000となる。上位29ビットについては、命令2601の packets アドレス29' h000000000から命令2606の packets アドレス29' h000000001を減算して、結果は29' h1fffffとなる。よって、その2つの演算結果を結合したPC相対値は、13ビットで表現可能であり、13' h1fff8となる。

【0139】

この方式によると、下位ビットの加算を省略することが可能となる。

(リニアアドレス方式)

リニアアドレス方式は、他の方式と違い、アドレスを分離せずに計算を行う。すなわち、リンカ、アセンブラ、コンパイラにおいては、上位ビット、下位ビットに分離せず、32ビットの通常の減算によりPC相対値を求め、プロセッサにおいても、上位ビット、下位ビットに分離せず、32ビットの通常の加算により分岐先アドレスを求める。

【0140】

この方式の場合、前述のリンカ307を例にとると図27のオブジェクトコードが出力される。命令2706と命令2713のPC相対値が図21と異なっている。命令2706のPC相対値の算出法について説明する。下位3ビットについては、命令2701のアドレス32'h00000000から命令2706のアドレス32'h0000000aを減算した結果、32'hffffffffff6となる。よって、そのPC相対値は、13ビットで表現可能であり、13'h1ffff6となる。

【0141】

これによって、一般の加算と同一の演算器で加算が実現できる。

(7) 実施の形態2において、アセンブラコード2906は、マスクROM、フラッシュメモリ等の半導体集積メモリや、フロッピーディスク、ハードディスク等の磁気記録媒体や、CD-ROM, DVD等の光ディスクに記録することもできる。

(8) 実施の形態2において、図30に示した命令は定数値とレジスタの加減算であったが、これに限らない。レジスタ間の加減算とすることもできる。PCとレジスタの加減算であることもできる。PCと定数の加減算の結果をレジスタに転送することもできる。加減算は論理演算であることもできる。

(9) 実施の形態3において、図34に示したPC加算器3406の計算方式は実施の形態1の桁上がり方式に限らない。上記した分離方式、絶対値方式、リニア方式のいずれでも実現可能である。

【0142】

【発明の効果】

請求項1記載のプロセッサによって、バイト単位にアドレッシングされない長さの命令を実行するプロセッサが実現できる。

【0143】

請求項2記載のプロセッサによって、プログラムカウンタのインクリメントを単純にすることができる。

【0144】

請求項3記載のプロセッサによって、分岐命令を実現できる。

請求項4記載のプロセッサによって、並列実行可能なプロセッサの分岐命令を実現できる。

【0145】

請求項5記載のプロセッサによって、並列実行可能なプロセッサの分岐命令をスカラプロセッサと同様の記述により実現できる。

【0146】

請求項6又は請求項7記載のプロセッサによって、プログラムカウンタとプログラムカウンタ相対値の加算を単純なハードウェアで実現できる。

【0147】

請求項8記載のプロセッサによって、プログラムカウンタとプログラムカウンタ相対値の加算の、上位と下位の自然な連続性を表現できる。

【0148】

請求項9記載のプロセッサによって、下位の加算を省略できる。

請求項10記載のプロセッサによって、一般の数の加算と同一の演算器で加算が実現できる。

【0149】

請求項11記載のプロセッサによって、プログラムカウンタとプログラムカウンタ相対値の加算を単純なハードウェアで実現し、プログラムカウンタ相対分岐を実現できる。

【0150】

請求項12記載のリンカによって、単純な減算方法で、上記プロセッサに対応

するリンクを実現できる。

【0151】

請求項13記載のリンクによって、上位と下位の自然な連続性を表現できる減算方法で、上記プロセッサに対応するリンクを実現できる。

【0152】

請求項14記載のリンクによって、下位の減算を省略できる減算方法で、上記プロセッサに対応するリンクを実現できる。

【0153】

請求項15記載のアセンブラによって、単純な減算方法で、上記プロセッサに対応するアセンブラを実現できる。

【0154】

請求項16記載のアセンブラによって、上位と下位の自然な連続性を表現できる減算方法で、上記プロセッサに対応するアセンブラを実現できる。

【0155】

請求項17記載のアセンブラによって、下位の減算を省略できる減算方法で、上記プロセッサに対応するアセンブラを実現できる。

【0156】

請求項18記載の命令列最適化装置によって、単純な減算方法による予測で、上記プロセッサに対応する命令列最適化装置を実現できる。

【0157】

請求項19記載の命令列最適化装置によって、上位と下位の自然な連続性を表現できる減算方法による予測で、上記プロセッサに対応する命令列最適化装置を実現できる。

【0158】

請求項20記載の命令列最適化装置によって、下位の減算を省略できる減算方法による予測で、上記プロセッサに対応する命令列最適化装置を実現できる。

【0159】

請求項21記載の逆アセンブラ装置によって、プログラムカウンタとプログラムカウンタ相対値の加算を単純な計算で実現し、逆アセンブラ装置を実現できる

【0160】

請求項 22 記載の逆アセンブラ装置によって、プログラムカウンタとプログラムカウンタ相対値の加算の、上位と下位の自然な連続性を表現でき、逆アセンブラ装置を実現できる。

【0161】

請求項 23 記載の逆アセンブラ装置によって、下位の加算を省略でき、逆アセンブラ装置を実現できる。

【0162】

請求項 24 記載のコンパイラによって、プログラムカウンタとプログラムカウンタ相対値の加算を単純なハードウェアで実現するプログラムカウンタ相対分岐命令を命令コード中に実現できる。

【0163】

請求項 25 記載のコンパイラによって、プログラムカウンタとプログラムカウンタ相対値の加算の、上位と下位の自然な連続性を表現するプログラムカウンタ相対分岐を命令コード中に実現できる。

【0164】

請求項 26 記載のコンパイラによって、下位の加算を省略できるプログラムカウンタ相対分岐を命令コード中に実現できる。

【0165】

請求項 27 記載のデバッガによって、上記プロセッサのの命令をデバッグするデバッガが実現できる。

【0166】

請求項 28 から請求項 43 記載の記録媒体によって、プログラムの開発及び、プロセッサでの実行が実現できる。

【0167】

以上のように、本発明によって、バイト単位にアドレッシングされない長さの命令を実行するプロセッサ及びリンカ、アセンブラ、最適化装置、コンパイラ、デバッガが実現され、特に高性能かつ小ハードウェアのプロセッサとしてその実用

的価値は大きい。

【図面の簡単な説明】

【図 1】

本発明に係るプロセッサの命令フォーマットを示す図

【図 2】

本プロセッサにおける命令の供給と発行を示す概念図

【図 3】

本発明に係るプロセッサ、コンパイラ、命令列最適化装置、アセンブラ、リンカから構成されるブロック図

【図 4】

実施の形態 1 におけるプロセッサ 309 の詳細を示すブロック図

【図 5】

本プロセッサにおけるアドレスのインクリメント方法を示す図

【図 6】

アドレスの計算方式の一つである桁上がり方式における下位 3 ビットの加算表と減算表を示す図

【図 7】

リンカ 307 の構成及び関連する入出力データを示すブロック図

【図 8】

アセンブラ 305 の構成及び関連する入出力データを示すブロック図

【図 9】

最適化装置 303 の構成及び入出力データを示すブロック図

【図 10】

最適化装置 303 のコード最適化手段において、コンパイラ 301 が出力したアセンブラコード 302 を最適化した最適化処理コード 903 の例を示す図

【図 11】

アドレス付与手段 904 において、図 9 のアセンブラコードにアドレスが付与されたことを示す図

【図 12】

ラベル検出手段 905 において出力したラベル情報 906 の例を示す図

【図 13】

最適化装置 303 が出力した最適化コードの例を示す図

【図 14】

機械語コード生成手段 802 において、図 13 の最適化コードに仮のアドレスを割り当てたことを示す図

【図 15】

ラベル検出手段 804 において出力したラベル情報 805 を示す図

【図 16】

アセンブラ 305 が出力したリロケータブルコード 814 の例を示す図

【図 17】

アセンブラ 305 が出力したリロケータブルコード 814 の例を示す図

【図 18】

コード結合手段 702 において、図 16 と図 17 のリロケータブルコードを結合し、実アドレスを与えた結合コードを示す図

【図 19】

コード結合手段 702 において、図 18 の命令 1810 中のラベルを解決したことを示す図

【図 20】

再配置情報検出手段 704 において出力した再配置情報 705 を示す図

【図 21】

再配置情報解決手段 713 において、未解決であった再配置情報を解決したことを示す図

【図 22】

分岐命令の PC 相対値を実行単位の先頭アドレスから分岐先アドレスへの差分値とした場合に、リンカ 307 が出力するオブジェクトコード 714 を示す図

【図 23】

アドレスの計算方法の一つである分離方式における下位 3 ビットの加算表と減算表を示す図

【図 24】

同方式の場合にリンカ 307 が出力するオブジェクトコード 714 を示す図

【図 25】

アドレスの計算方法の一つである絶対位置指定方式における下位 3 ビットの加算表と減算表を示す図

【図 26】

同方式の場合にリンカ 307 が出力するオブジェクトコード 714 を示す図

【図 27】

アドレスの計算方法の一つであるリニアアドレス方式の場合に、リンカ 307 が出力するオブジェクトコード 714 を示す図

【図 28】

実施の形態 2 におけるプロセッサ 309 の詳細を示す図

【図 29】

実施の形態 2 におけるコンパイラ 301 の詳細を示すブロック図

【図 30】

実施の形態 2 におけるプロセッサのプログラムカウンタ加算命令とプログラムカウンタ減算命令を示した図

【図 31】

C 言語で記述されたプログラムを示す図

【図 32】

実施の形態 2 におけるコンパイラの間接コードの例を示す図

【図 33】

実施の形態 2 におけるコンパイラが出力したアセンブラコードを示す図

【図 34】

実施の形態 3 におけるデバッガ及び逆アセンブラの構成図

【図 35】

従来技術の命令及び命令パケットのフォーマットを示す図

【符号の説明】

100 並列実行境界情報

- 101 フォーマット情報
- 300 ソースコード
- 301 コンパイラ
- 302 アセンブラコード
- 303 最適化装置
- 304 最適化コード
- 305 アセンブラ
- 306 リロケータブルコード
- 307 リンカ
- 308 オブジェクトコード
- 309 プロセッサ
- 401 a 第1演算器
- 401 b 第2演算器
- 401 c 第3演算器
- 402 汎用レジスタ
- 403 上位PC
- 404 下位PC
- 405 PC演算器
- 406 データメモリ
- 407 命令メモリ
- 408 命令バッファ
- 409 a 第1命令デコーダ
- 409 b 第2命令デコーダ
- 409 c 第3命令デコーダ
- 410 プリフェッチカウンタ
- 701 リロケータブルコード
- 702 コード結合手段
- 703 結合コード
- 704 再配置情報検出手段

705 再配置情報
 706 下位アドレス減算手段
 707 桁借り数
 708 下位減算結果
 709 上位アドレス減算手段
 710 上位減算結果
 711 アドレス差算出手段
 712 アドレス差
 713 再配置情報解決手段
 714 オブジェクトコード
 801 最適化コード
 802 機械語コード生成手段
 803 機械語コード
 804 ラベル検出手段
 805 ラベル情報
 806 下位アドレス減算手段
 807 桁借り数
 808 下位減算結果
 809 上位アドレス減算手段
 810 上位減算結果
 811 アドレス差算出手段
 812 アドレス差
 813 ラベル情報解決手段
 814 リロケータブルコード
 901 アセンブラコード
 902 コード最適化手段
 903 最適化処理コード
 904 アドレス付与手段
 905 ラベル検出手段

906 ラベル情報
 907 下位アドレス減算手段
 908 桁借り数
 909 下位減算結果
 910 上位アドレス減算手段
 911 上位減算結果
 912 アドレス差算出手段
 913 アドレス差
 914 ラベル情報解決手段
 915 最適化コード
 916 アドレス付与コード
 1000～1008 最適化されたニーモニック命令
 1100～1108 アドレスが付与された命令
 1300～1303 命令が連結されたパケット
 1400 パケット
 1401～1403 パケット1400中の命令
 1404 パケット
 1405、1406 パケット1404中の命令
 1407 パケット
 1408～1410 パケット1407中の命令
 1411 パケット
 1600 パケット
 1601～1603 パケット1600中の命令
 1604 パケット
 1605、1606 パケット1604中の命令
 1607 パケット
 1608～1610 パケット1607中の命令
 1611 パケット
 1700 パケット

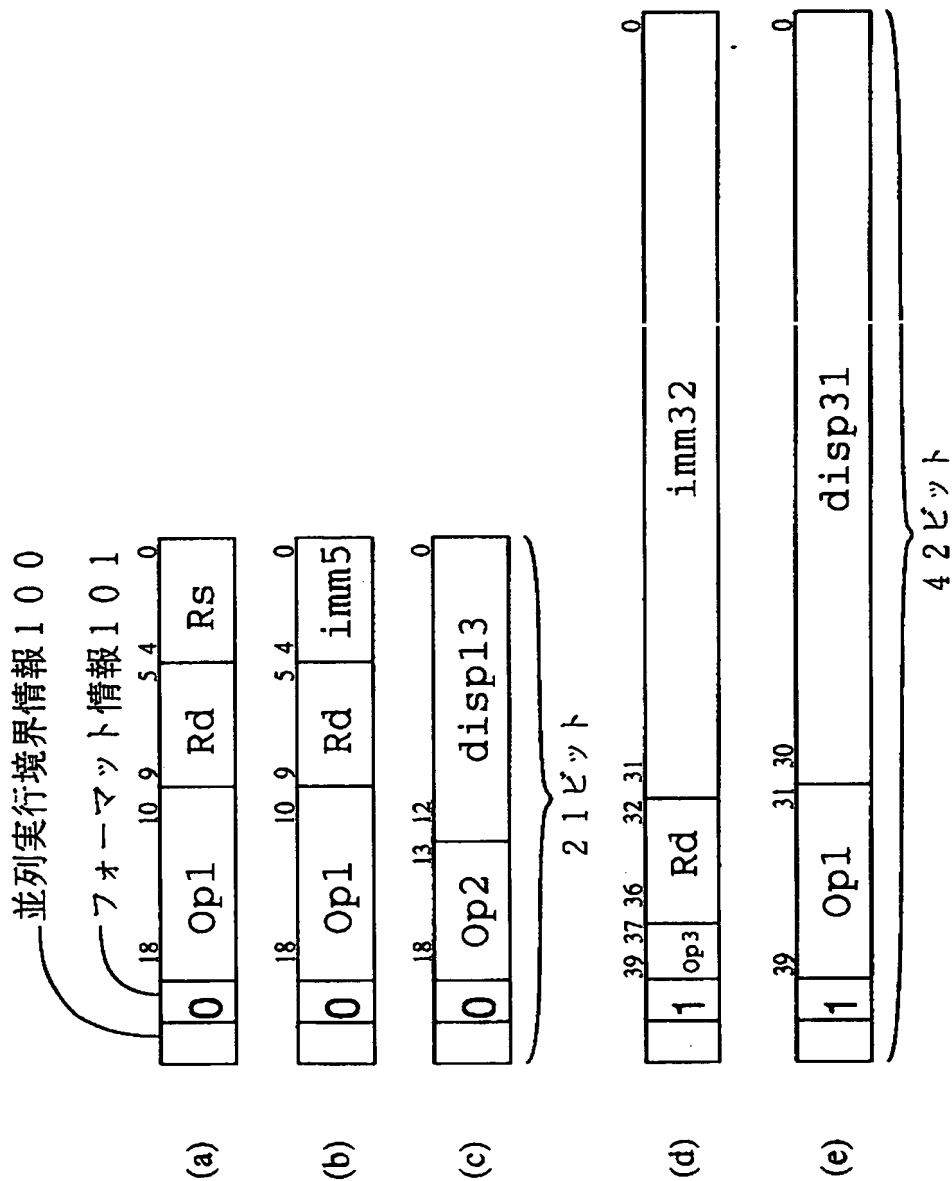
1701~1703 パケット1700中の命令
1800 パケット
1801~1803 パケット1800中の命令
1804 パケット
1805~1807 パケット1804中の命令
1808 パケット
1809、1810 パケット1808中の命令
1811 パケット
1812~1814 パケット1811中の命令
1815 パケット
1900 パケット
1901~1903 パケット1900中の命令
1904 パケット
1905~1907 パケット1904中の命令
1908 パケット
1909、1910 パケット1908中の命令
1911 パケット
1912~1914 パケット1911中の命令
1915 パケット
2100 パケット
2101~2103 パケット2100中の命令
2104 パケット
2105~2107 パケット2104中の命令
2108 パケット
2109、2110 パケット2108中の命令
2111 パケット
2112~2114 パケット2111中の命令
2115 パケット
2200 パケット

2201～2203 パケット2200中の命令
2204 パケット
2205～2207 パケット2204中の命令
2208 パケット
2209、2210 パケット2208中の命令
2211 パケット
2212～2214 パケット2211中の命令
2215 パケット
2400 パケット
2401～2403 パケット2400中の命令
2404 パケット
2405～2407 パケット2404中の命令
2408 パケット
2409、2410 パケット2408中の命令
2411 パケット
2412～2414 パケット2411中の命令
2415 パケット
2600 パケット
2601～2603 パケット2600中の命令
2604 パケット
2605～2607 パケット2604中の命令
2608 パケット
2609、2610 パケット2608中の命令
2611 パケット
2612～2614 パケット2611中の命令
2615 パケット
2700 パケット
2701～2703 パケット2700中の命令
2704 パケット

2705～2707 パケット2704中の命令
2708 パケット
2709、2710 パケット2708中の命令
2711 パケット
2712～2714 パケット2711中の命令
2715 パケット
2801 第2PC演算器
2901 ソースコード
2902 中間コード変換部
2903 中間コード
2904 PC値加算命令変換部
2905 命令変換部
2906 アセンブラコード
3201～3215 中間コード
3301～3316 アセンブラコード
3401 命令メモリ
3402 マイクロプロセッサ
3403 ラベルテーブル
3404 命令バッファ
3405 表示部
3406 PC加算器
3407 制御部

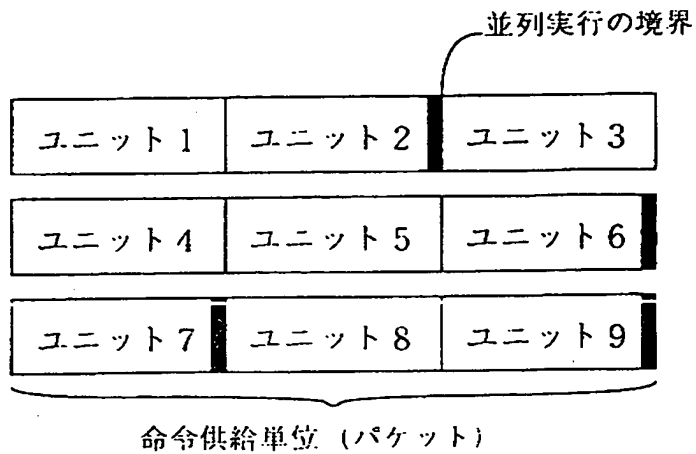
【書類名】 図面

【図 1】



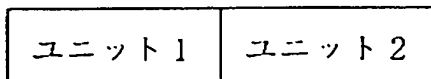
【図2】

(a) 実行コード

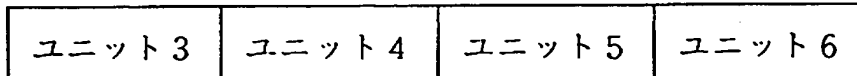


(b) 実行イメージ

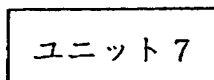
実行単位 1



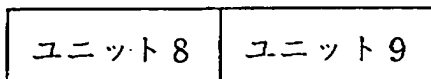
実行単位 2



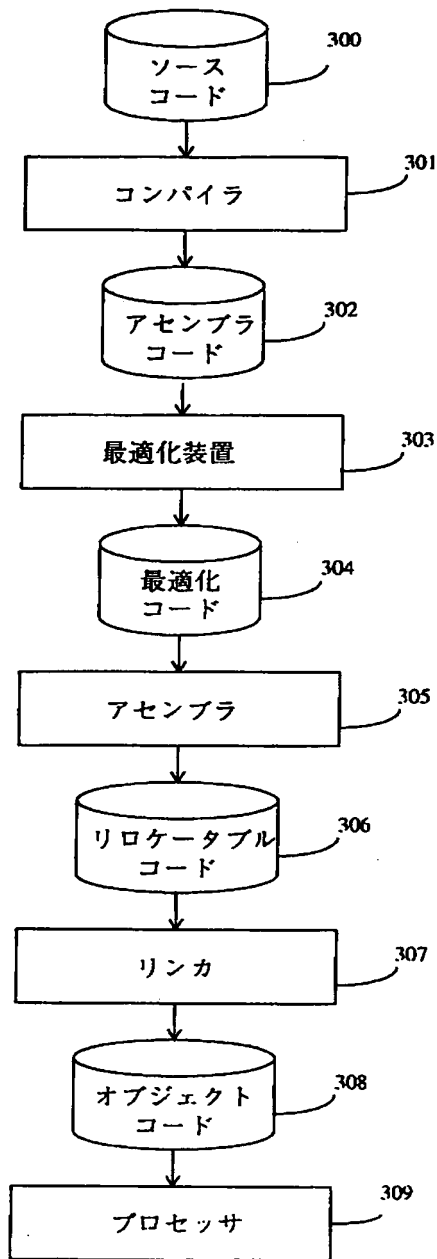
実行単位 3



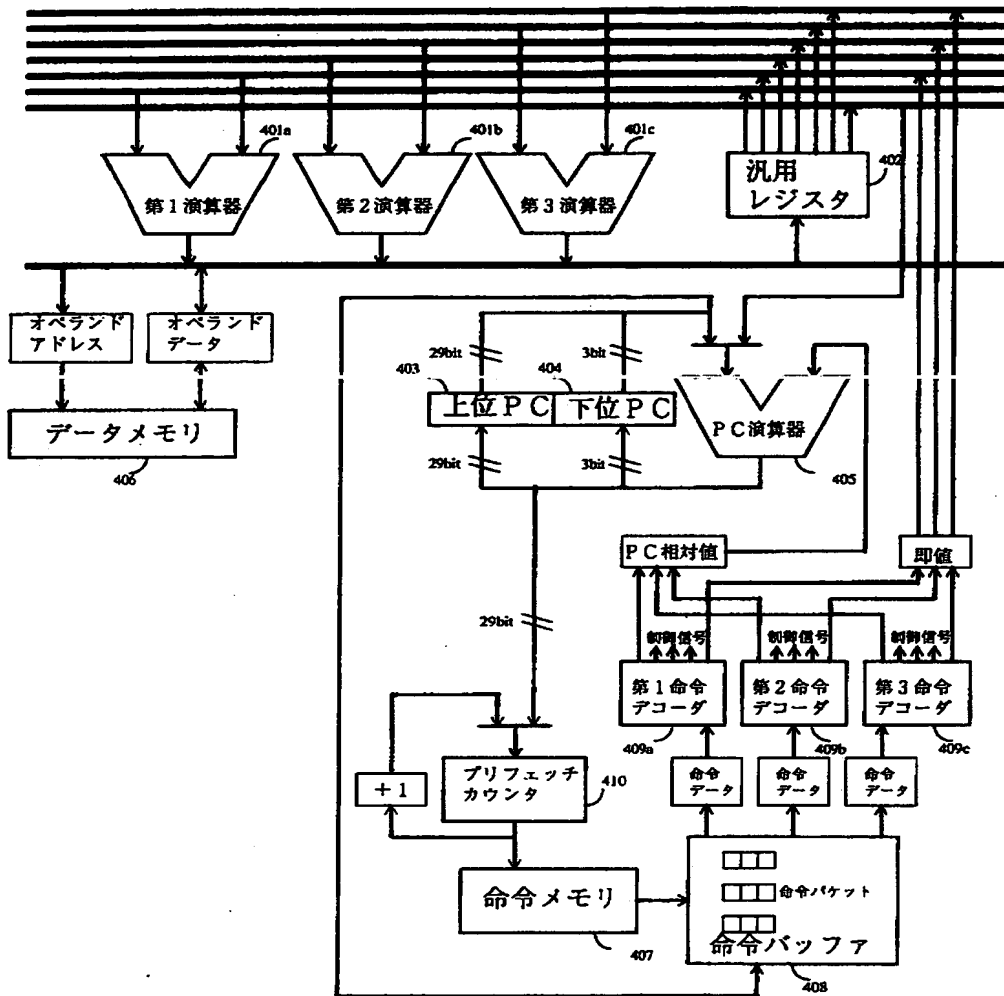
実行単位 4



【図 3】



【図 4】



【図 5】

インクリメント表

更新前の バケット内 アドレス インクリメント値	3'b000	3'b010	3'b100
1	3'b010	3'b100	3'b000 (桁上がり 1)
2	3'b100	3'b000 (桁上がり 1)	3'b010 (桁上がり 1)
3	3'b000 (桁上がり 1)	3'b010 (桁上がり 1)	3'b100 (桁上がり 1)
4	3'b010 (桁上がり 1)	3'b100 (桁上がり 1)	3'b000 (桁上がり 2)

【図6】

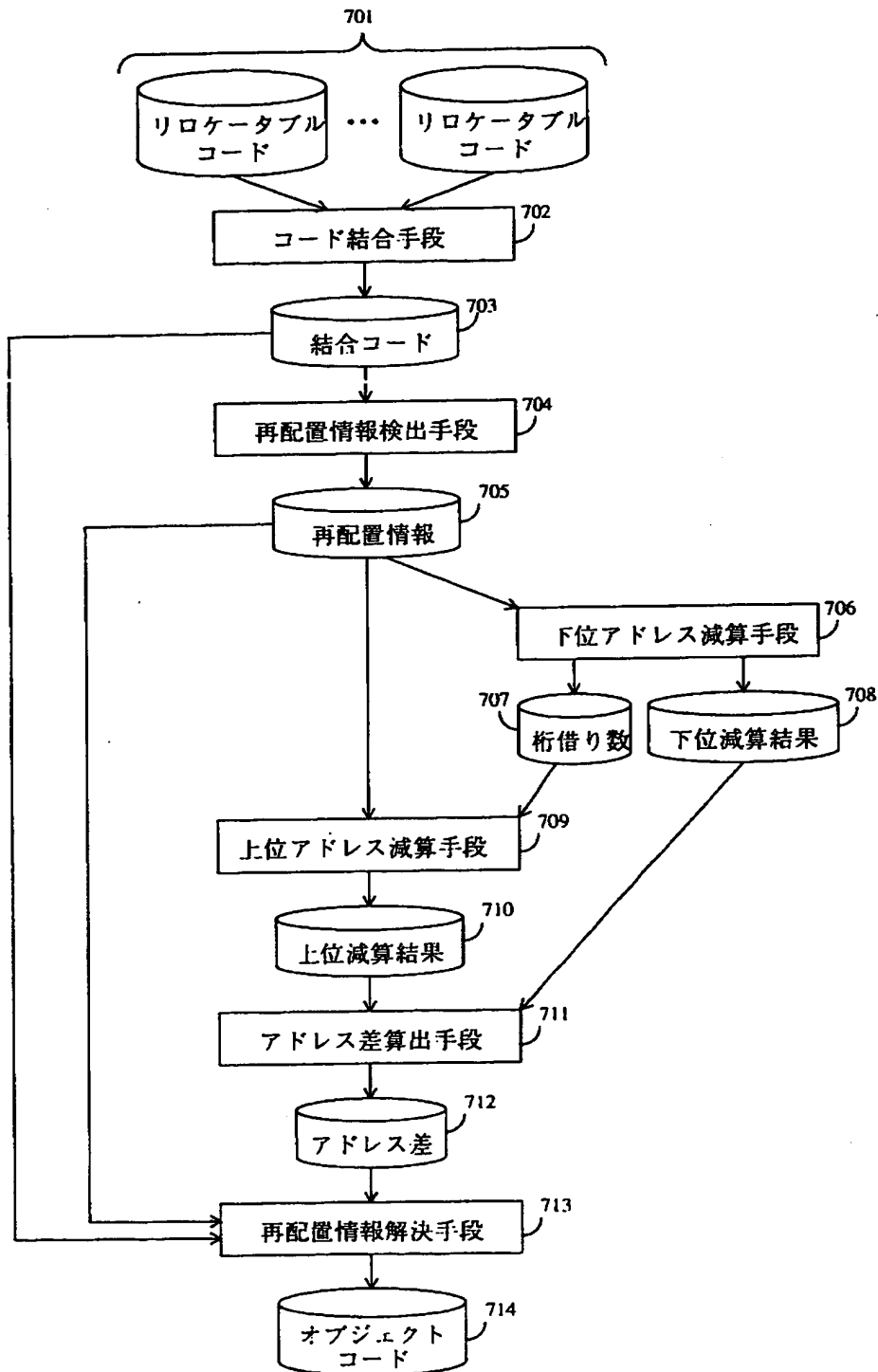
(a) 加算表

アドレス値の 下位3ビット PC 相対値の 下位3ビット	アドレス値の 下位3ビット		
	0b000	0b010	0b100
0b000	0b000	0b010	0b100
0b010	0b010	0b100	0b000 (桁上がり 1)
0b100	0b100	0b000 (桁上がり 1)	0b010 (桁上がり 1)

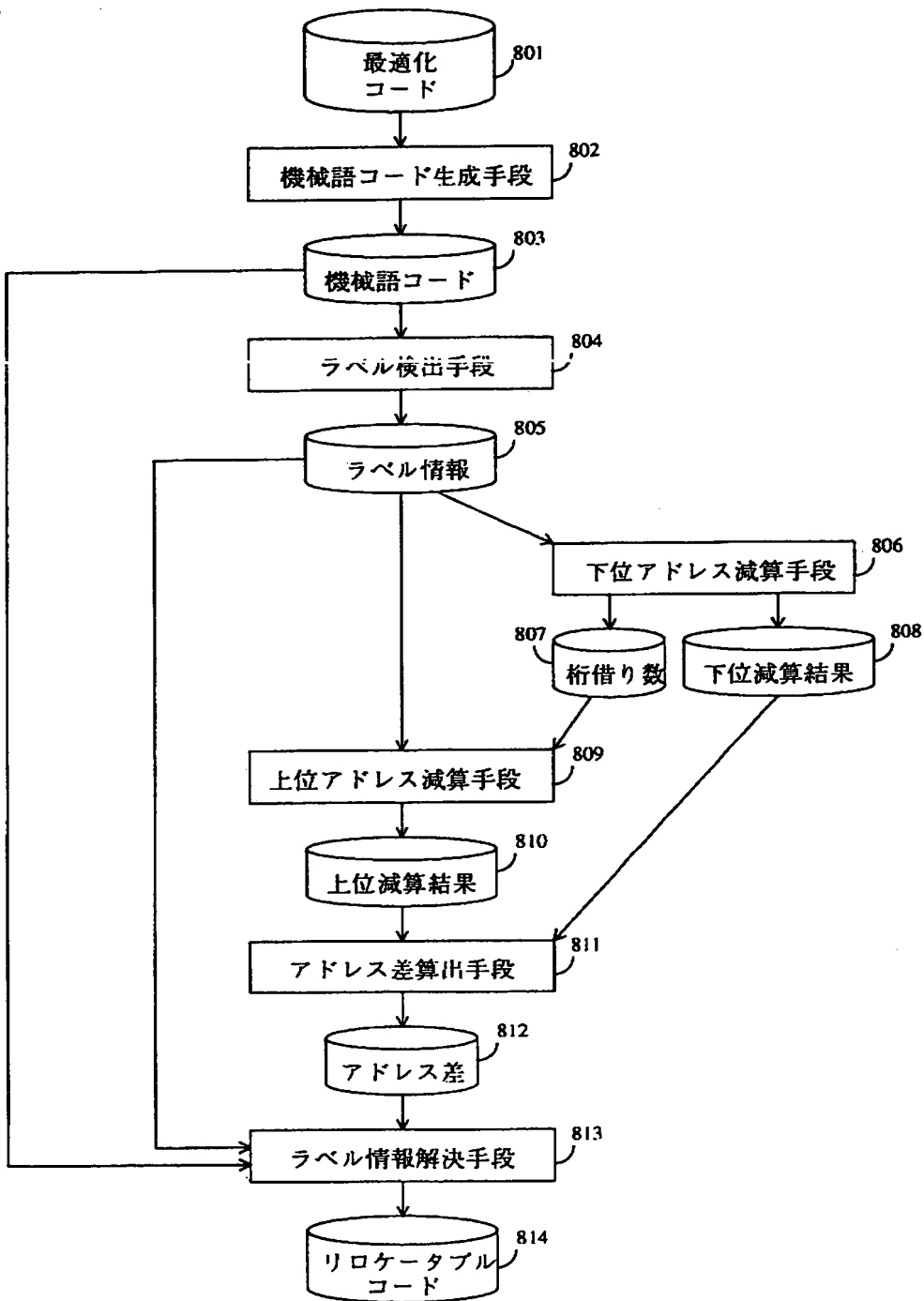
(b) 減算表

アドレス値の 下位3ビット (減算される数)	(減算する数) アドレス値の 下位3ビット		
	0b000	0b010	0b100
0b000	0b000	0b100 (桁借り 1)	0b010 (桁借り 1)
0b010	0b010	0b000	0b100 (桁借り 1)
0b100	0b100	0b010	0b000

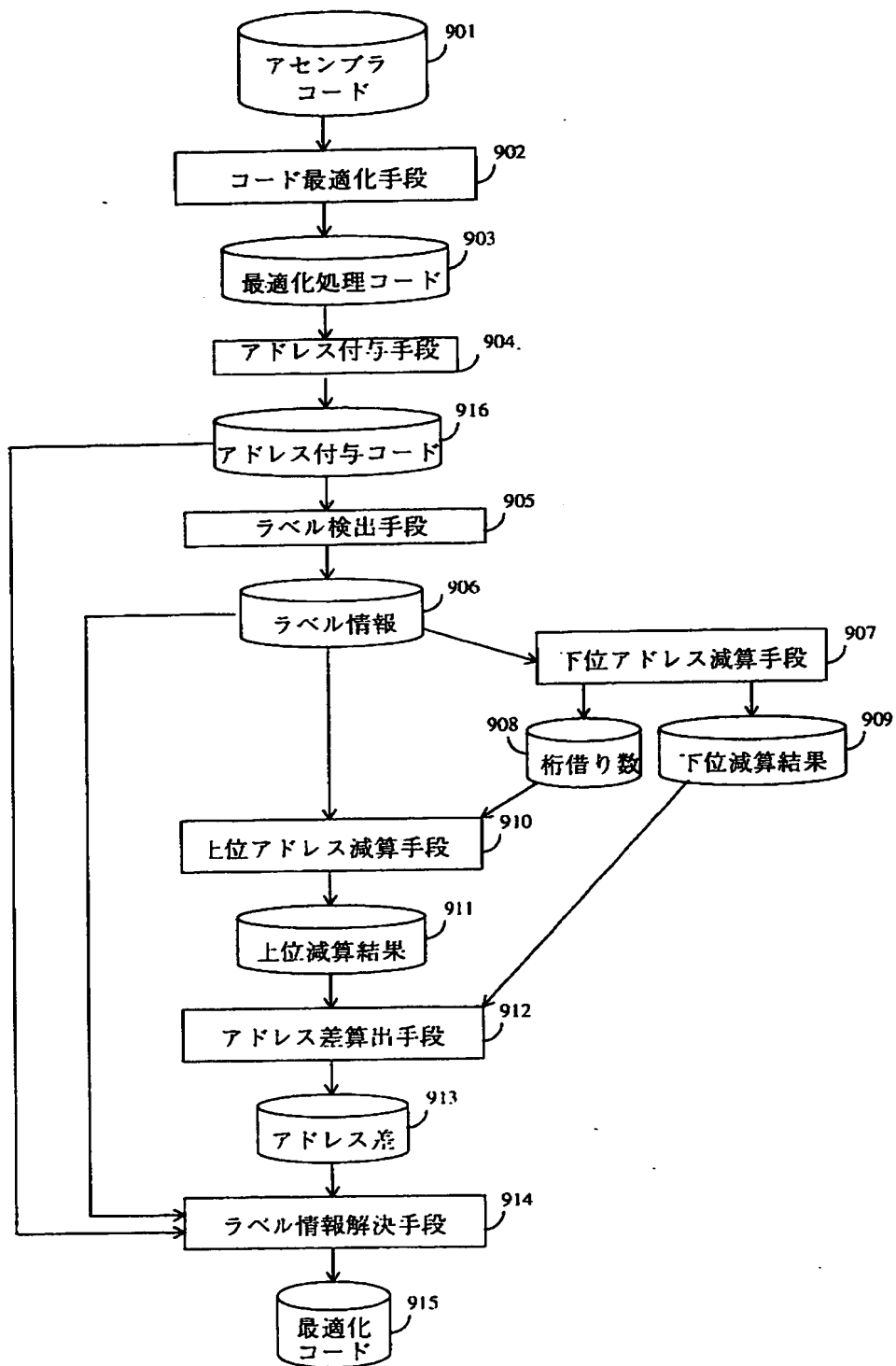
【図 7】



【図 8】



【図 9】



【図 1 0】

L1:	mov r2,r1	・ ・	1000
	jsr f	・ ・	1001
	add r0,r4	・ ・	1002
	and r1,r3	・ ・	1003
	mov L2,r2	・ ・	1004
	ld (r2),r0	・ ・	1005
	bra L1	・ ・	1006
	add r2,r3	・ ・	1007
	...		
L2:	...	・ ・	1008

【図 1 1】

32'h00000800	L1: mov r2,r1	・ ・ ・ ・	1100
32'h00000802	jsr f	・ ・ ・ ・	1101
32'h00000804	add r0,r4	・ ・ ・ ・	1102
32'h00000808	and r1,r3	・ ・ ・ ・	1103
32'h0000080a	mov L2,r2	・ ・ ・ ・	1104
32'h00000810	ld (r2),r0	・ ・ ・ ・	1105
32'h00000812	bra L1	・ ・ ・ ・	1106
32'h00000814	add r2,r3	・ ・ ・ ・	1107
	...		
32'h12345678	L2: ...	・ ・ ・ ・	1108

【図 12】

命令	解決する値
mov L2,r2	アドレス 32'h12345678
bra L1	P C 相対値 32'h00000800-32'h00000812

【図 13】

L1:	mov r2,r1		jsr f		add r0,r4	・ ・ 1300
	and r1,r3		mov L2,r2		(mov L2,r2)	・ ・ 1301
	ld (r2),r0		bra L1		add r2,r3	・ ・ 1302
	...					
L2:						・ ・ 1303

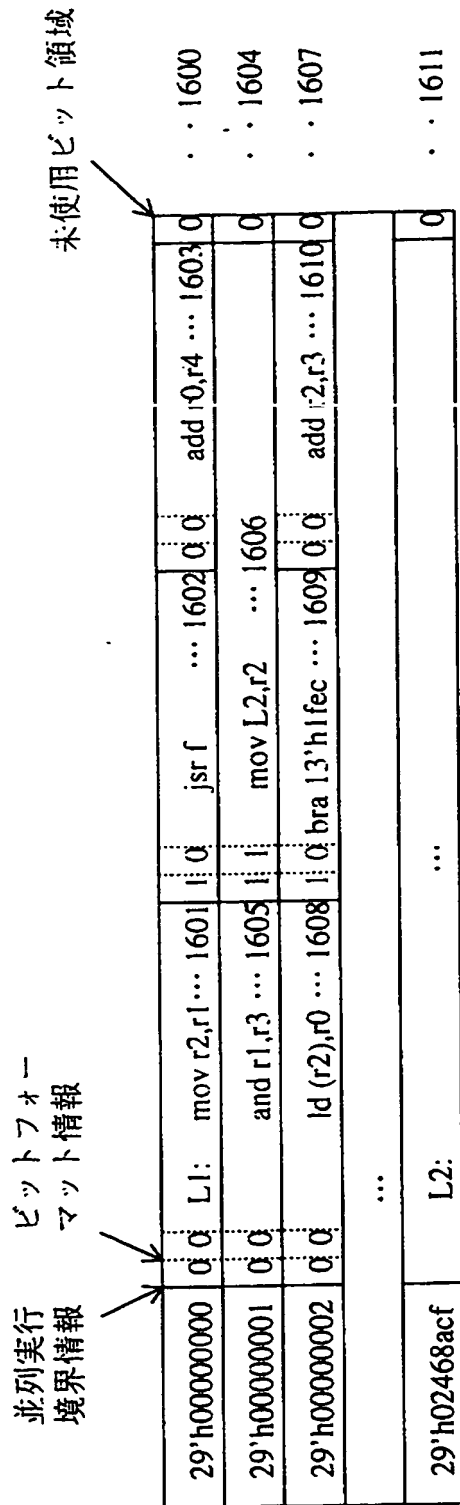
【図 1 4】

29'h00000000	L1: mov r2,r1... 1401	jsr f ... 1402	add r0,r4 ... 1403	. . 1400
29'h00000001	and r1,r3 ... 1405	mov L2,r2 ... 1406		. . 1404
29'h00000002	ld (r2),r0 ... 1408	bra L1 ... 1409	add r2,r3 ... 1410	. . 1407
...				
29'h02468acf	L2: 1411

【図 15】

命令	解決する値
bra L1	P C 相対値 32'h00000000-32'h00000812

【図 16】



【図 1 7】

29'h00000000	0 0	f:	ret	...	1701	0 0	nop	...	1702	1 0	nop	...	1703	0	...	1700
--------------	-----	----	-----	-----	------	-----	-----	-----	------	-----	-----	-----	------	---	-----	------

【図 1 8】

29'h00000000	0 0	f:	ret	... 1801	0 0	nop	... 1802	1 0	nop	... 1803	0	.. 1800
29'h00000001	0 0	L1:	mov r2,r1	... 1805	1 0	jsr f	... 1806	0 0	add r0,r4	... 1807	0	.. 1804
29'h00000002	0 0		and r1,r3	... 1809	1 1	mov L2,r2	... 1810				0	.. 1808
29'h00000003	0 0		ld (r2),r0	... 1812	1 0	bra 0x1fec	... 1813	0 0	add r2,r3	... 1814	0	.. 1811
...												
29'h02468ad0		L2:				...					0	.. 1815

【図 19】

29'h00000000	0 0	f:	ret	... 1901	0 0	nop	... 1902	1 0	nop	... 1903	0	.. 1900
29'h00000001	0 0	L1:	mov r2,r1	... 1905	1 0	jsr f	... 1906	0 0	add r0,r4	... 1907	0	.. 1904
29'h00000002	0 0		and r1,r3	... 1909	1 1	mov 32'h12345680,r2	... 1910				0	.. 1908
29'h00000003	0 0		ld (r2),r0	... 1912	1 0	bra 0x1fec	... 1913	0 0	add r2,r3	... 1914	0	.. 1911
...												
29'h02468ad0		L2:				...					0	.. 1915

【図 20】

命令	解決する値
jsr f	P C 相対値 32'h00000000-32'h0000000a

【図 21】

29'h00000000	0 0	f:	ret	... 2101	0 0	nop	... 2102	1 0	nop	... 2103	0	. . 2100
29'h00000001	0 0	L1:	mov r2,r1	... 2105	1 0	jsr 13'h1fff4	... 2106	0 0	add r0,r4	... 2107	0	. . 2104
29'h00000002	0 0		and r1,r3	... 2109	1 1	mov 32'h12345680,r2	... 2110	0			0	. . 2108
29'h00000003	0 0		ld (r2),r0	... 2112	1 0	bra 13'h1fec	... 2113	0 0	add r2,r3	... 2114	0	. . 2111
...												
29'h02468ad0		L2:				...					0	. . 2115

【図 2 2】

29'h00000000	0 0	f:	ret	... 2201	0 0	nop	... 2202	1 0	nop	... 2203	0	. . 2200
29'h00000001	0 0	L1:	mov r2,r1	... 2205	1 0	jsr 13'h1fff8	... 2206	0 0	add r0,r4	... 2207	0	. . 2204
29'h00000002	0 0		and r1,r3	... 2209	1 1	mov 32'h12345680,r2	... 2210				0	. . 2208
29'h00000003	0 0		ld (r2),r0	... 2212	1 0	bra 13'h1fff0	... 2213	0 0	add r2,r3	... 2214	0	. . 2211
...												
29'h02468ad0		L2:				...					0	. . 2215

【図 23】

(a) 加算表

アドレス値の 下位 3 ビット PC 相対値の 下位 3 ビット	アドレス値の 下位 3 ビット		
	0b000	0b010	0b100
0b000	0b000	0b010	0b100
0b010	0b010	0b100	0b000 (桁上がり無視)
0b100	0b100	0b000 (桁上がり無視)	0b010 (桁上がり無視)

(b) 減算表

アドレス値の 下位 3 ビット (減算される数)	(減算する数) アドレス値の 下位 3 ビット		
	0b000	0b010	0b100
0b000	0b000	0b100 (桁借り無視)	0b010 (桁借り無視)
0b010	0b010	0b000	0b100 (桁借り無視)
0b100	0b100	0b010	0b000

【図 2 4】

29'h00000000	0 0	f:	ret	... 2401	0 0	nop	... 2402	1 0	nop	... 2403	0	. . 2400
29'h00000001	0 0	L1:	mov r2,r1	... 2405	1 0	jsr 13'h1ffc	... 2406	0 0	add r0,r4	... 2407	0	. . 2404
29'h00000002	0 0		and r1,r3	... 2409	1 1	mov 32'h12345680,r2	... 2410				0	. . 2408
29'h00000003	0 0		ld (r2),r0	... 2412	1 0	bra 13'h1ff4	... 2413	0 0	add r2,r3	... 2414	0	. . 2411
...												
29'h02468ad0		L2:		...							0	. . 2415

【図 25】

(a) 加算表

アドレス値の 下位 3 ビット PC 相対値の 下位 3 ビット	アドレス値の 下位 3 ビット		
	0b000	0b010	0b100
0b000	0b000	0b000	0b000
0b010	0b010	0b010	0b010
0b100	0b100	0b100	0b100

(b) 減算表

アドレス値の 下位 3 ビット (減算される数)	(減算する数) アドレス値の 下位 3 ビット		
	0b000	0b010	0b100
0b000	0b000	0b000	0b000
0b010	0b010	0b010	0b010
0b100	0b100	0b100	0b100

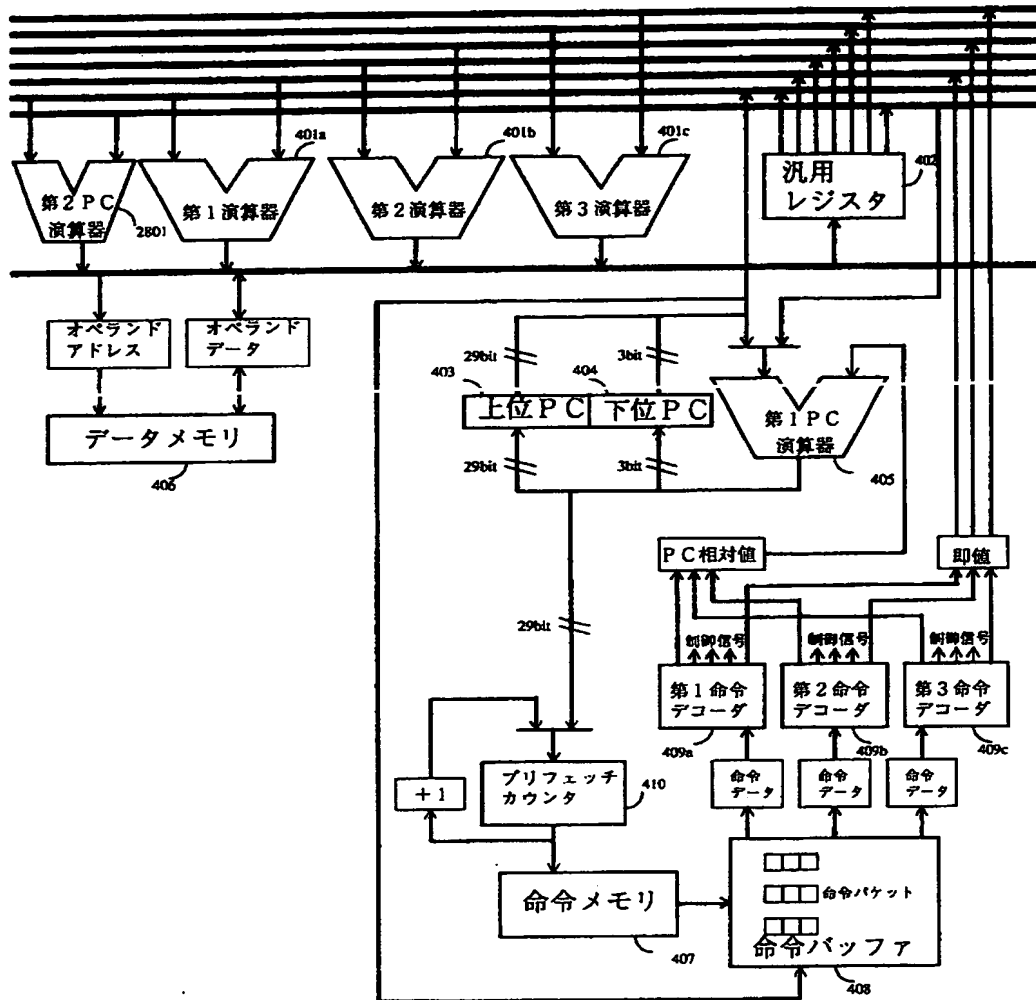
【図 2 6】

29'h00000000	0 0	f:	ret	... 2601	0 0	nop	... 2602	1 0	iop	... 2603	0	. . 2600
29'h00000001	0 0	L1:	mov r2,r1	... 2605	1 0	jsr 13'h1fff8	... 2606	0 0	add r0,r4	... 2607	0	. . 2604
29'h00000002	0 0		and r1,r3	... 2609	1 1	mov 32'h12345680,r2	... 2610				0	. . 2608
29'h00000003	0 0		ld (r2),r0	... 2612	1 0	bra 13'h1fff0	... 2613	0 0	add r2,r3	... 2614	0	. . 2611
...												
29'h02468ad0		L2:				...					0	. . 2615

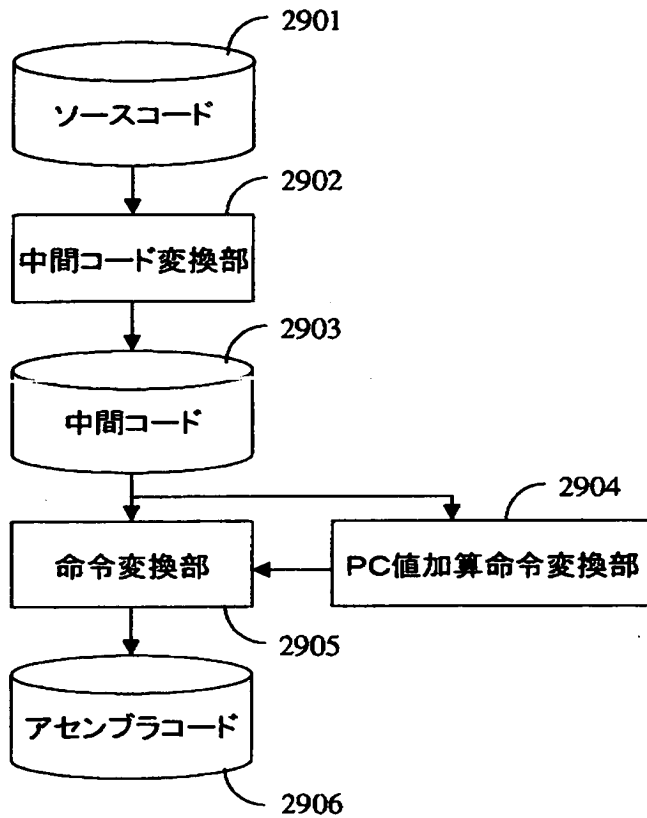
【図 2 7】

29'h00000000	0 0	f:	ret	... 2701	0 0	nop	... 2702	1 0	nop	... 2703	0	. . 2700
29'h00000001	0 0	L1:	mov r2,r1	... 2705	1 0	jsr l3'h1fff6	... 2706	0 0	add r0,r4	... 2707	0	. . 2704
29'h00000002	0 0		and r1,r3	... 2709	1 1	mov 32'h12345680,r2	... 2710				0	. . 2708
29'h00000003	0 0		ld (r2),r0	... 2712	1 0	bra l3'h1fee	... 2713	0 0	add r2,r3	... 2714	0	. . 2711
...												
29'h02468ad0		L2: 2715
											0	

【図 28】



【図 29】



【図 30】

	ニモニック	オペレーション
(a)	addpc disp,Rn	$Rn + \text{disp} \rightarrow Rn$
(b)	subpc disp,Rn	$Rn - \text{disp} \rightarrow Rn$

【図31】

```
extern int g1();
extern int g2();
extern int g3();
extern int g4();

f(int i)
{
    int (*fp)();

    switch(i) {
        case 1: fp = g1;
                break;
        case 2: fp = g2;
                break;
        case 3: fp = g3;
                break;
        default: fp = g4;
    }

    (*fp)();
}
```

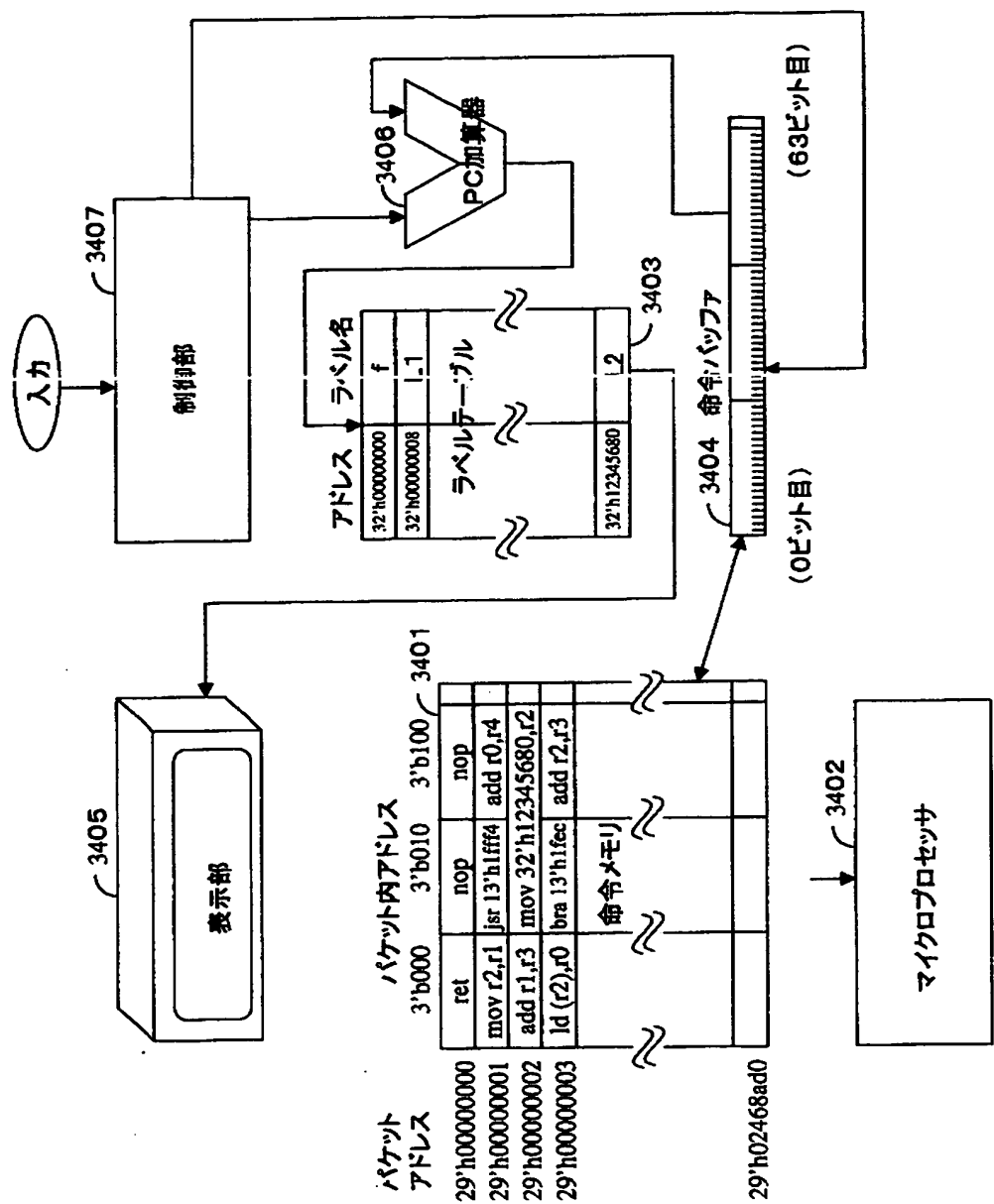
【図 3 2】

f:	tmp = PC	3 2 0 1
	i != 1	3 2 0 2
	br L1	3 2 0 3
	fp = (g1 - f) + tmp	3 2 0 4
	jmp L	3 2 0 5
L1:	i != 2	3 2 0 6
	br L2	3 2 0 7
	fp = (g2 - f) + tmp	3 2 0 8
	jmp L	3 2 0 9
L2:	i != 3	3 2 1 0
	br L3	3 2 1 1
	fp = (g3 - f) + tmp	3 2 1 2
	jmp L	3 2 1 3
L3:	fp = (g4 - f) + tmp	3 2 1 4
L:	*(fp)(i)	3 2 1 5

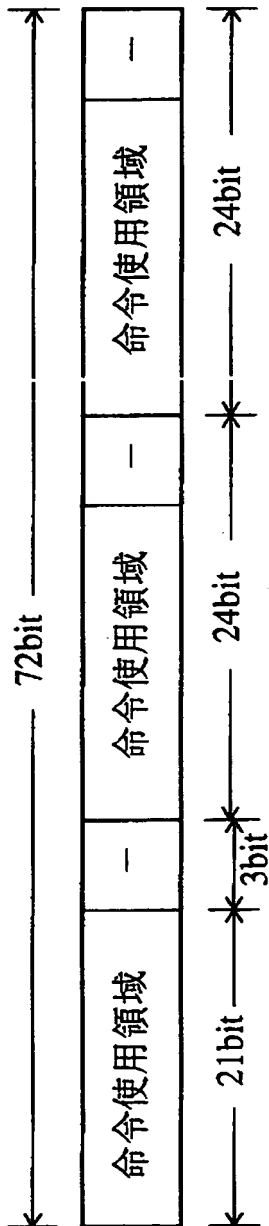
【図 3 3】

f:	mov	PC,r1	3 3 0 1
	cmpne	1,r0	3 3 0 2
	br	L1	3 3 0 3
	addpc	g1-f,r1	3 3 0 4
	jmp	L	3 3 0 5
L1:	cmpne	2,r0	3 3 0 6
	br	L2	3 3 0 7
	addpc	g2-f,r1	3 3 0 8
	jmp	L	3 3 0 9
L2:	cmpne	3,r0	3 3 1 0
	br	L3	3 3 1 1
	addpc	g3-f,r1	3 3 1 2
	jmp	L	3 3 1 3
L3:	addpc	g4-f,r1	3 3 1 4
L:	jsr	(r1)	3 3 1 5
	ret		3 3 1 6

【図 3 4】



【図 3 5】



【書類名】 要約書

【要約】

【課題】 バイト単位にアドレッシングされない長さの命令を実行するプロセッサを提供する。

【解決手段】 各種演算を行う演算器401a~401c、データ、アドレス等を格納する汎用レジスタ402、次に実行する命令の先頭アドレスを保持する上位プログラムカウンタ403と下位プログラムカウンタ404、上位PCと下位PCの各種演算を行うPC演算器405、データを記憶するデータメモリ406と命令を記憶する命令メモリ407、さらに命令メモリ407からフェッチした命令を格納する命令バッファ408、命令バッファ内の実行可能な命令をそれぞれデコードする命令デコーダ409a~409cを備える。

【選択図】 図4

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000005821

【住所又は居所】 大阪府門真市大字門真 1006 番地

【氏名又は名称】 松下電器産業株式会社

【代理人】 申請人

【識別番号】 100078204

【住所又は居所】 大阪府門真市大字門真 1006 松下電器産業株式
会社内

【氏名又は名称】 滝本 智之

【選任した代理人】

【識別番号】 100097445

【住所又は居所】 大阪府門真市大字門真 1006 番地 松下電器産業
株式会社内

【氏名又は名称】 岩橋 文雄

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日

[変更理由] 新規登録

住 所 大阪府門真市大字門真1006番地

氏 名 松下電器産業株式会社